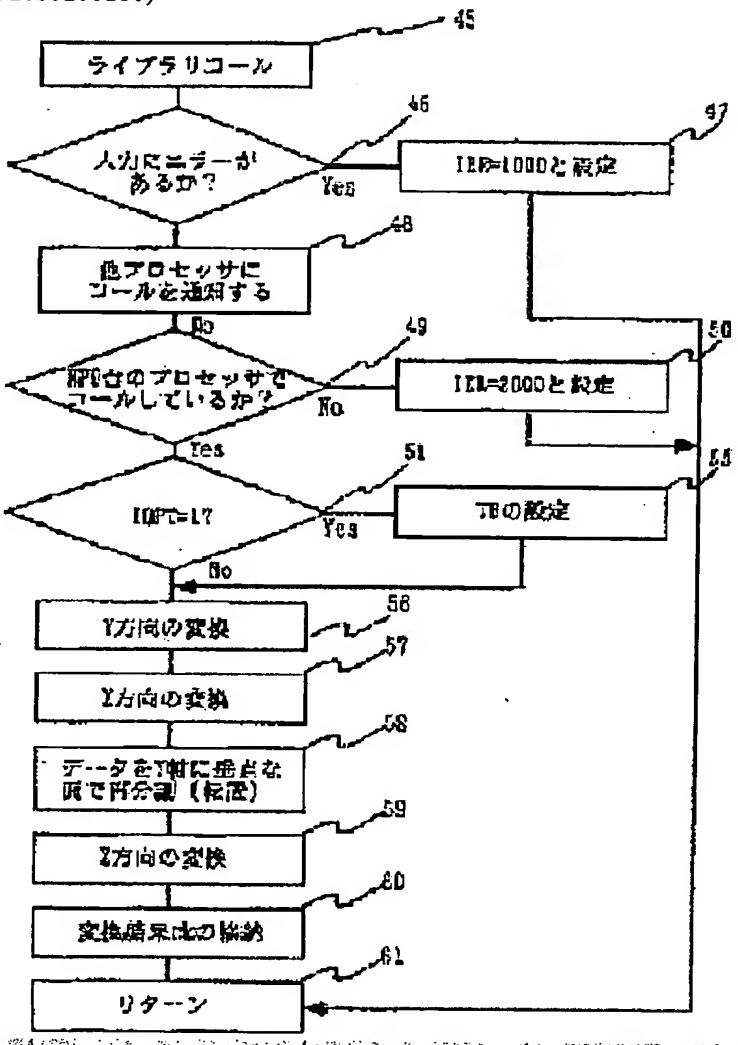
MicroPatent® PatSearch FullText: Record 1 of 1

Search scope: JP (bibliographic data only)
Years: 1991-2004
Patent/Publication No.: (JP2000200261)



JP2000200261 A
FOURIER TRANSFORMING METHOD, SIMULATION METHOD, AND PROGRAM RECORDING MEDIUM
HITACHI LTD
Inventor(s): YAMAMOTO YUSAKU ;NAONO TAKESHI
Application No. 10377684 JP10377684 JP, **Filed** 19981229,**A1 Published** 20000718**Published** 20000718

Abstract: PROBLEM TO BE SOLVED: To perform Fourier transformation fast on parallel computers while holding transformation object data and conversion result data in the same data division format.

SOLUTION: Linear data to be transformed are arranged in a rectangular parallelepiped shape and mapped into three-dimensional transformation object data, the rectangular parallelepiped is divided by a plane perpendicular to Z direction, and data of each plane are assigned to one processor, and respective processors perform Y-directional Fourier transformation for the three- dimensional transformation object data according to the allocation to obtain 1st

transformation result data (56). The respective processors perform X- directional Fourier transformation for the 1st transformation result data to obtain 2nd transformation result data (57). The 2nd transformation result data are divided by a plane perpendicular to the Y axis, the data are rearranged between the processors according to the divisions (58), and the respective processors perform transformation similar to Z-directional transformation to obtain final three-dimensional Fourier transformation result data (59).

Int'l Class: G06F01714; G06F01700

Patents Citing this One: No US, EP, or WO patents/search reports have cited this patent. **MicroPatent Reference Number:** 000555953

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号
特開2000－200261
（P2000－200261A）
(43)公開日　平成12年７月18日(2000.7.18)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　17/14		Ｇ０６Ｆ　15/332	Ａ	５Ｂ０４９
17/00		15/20	Ｄ	５Ｂ０５６

審査請求　未請求　請求項の数28　ＦＤ　（全 27 頁）

(21)出願番号　特願平10－377684

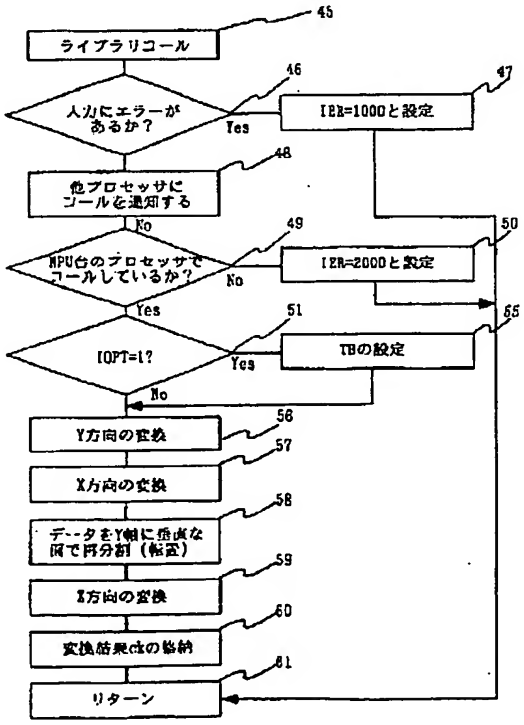
(22)出願日　平成10年12月29日(1998.12.29)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　山本　有作
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　直野　健
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内
(74)代理人　100061893
弁理士　高橋　明夫　（外１名）

Ｆターム(参考)　5B049 AA04 EE04 EE41
5B056 AA04 BB12

(54)【発明の名称】　フーリエ変換方法、シミュレーション方法およびプログラム記録媒体

(57)【要約】
【課題】並列計算機上でフーリエ変換を、変換対象データと変換結果データがデータ分割形式が同じに保ち、かつ、高速に実行する。
【解決手段】変換対象一次元データを直方体状に並べて３次元の変換対象データに写像し、この直方体をＺ方向に垂直な面で分割して各面のデータを一つのプロセッサに割り当て、各プロセッサで、この割り当てにしたがって３次元の変換対象データに対してＹ方向のフーリエ変換を実行して第１変換結果データを得る(56)。各プロセッサで、第１変換結果データに対してＸ方向のフーリエ変換に類似の変換を実行して第２変換結果データを得る(57)。第２変換結果データをＹ軸に垂直な面で分割し直し、その分割に従いそのデータをプロセッサ間で並び替え(58)、その後に、各プロセッサで、Ｚ方向のフーリエ変換に類似の変換を行い最終的な三次元フーリエ変換結果データを得る(59)。



【特許請求の範囲】
【請求項１】複数のプロセッサを有する計算機で実行するためのフーリエ変換方法であって、
各プロセッサにより、第１の変換処理、第２の変換処理、第３の変換処理を順次かつ他のプロセッサと並行して実行し、
上記複数のプロセッサの各々による、上記第１、第２の変換処理のいずれか一方の変換処理の実行後に、上記複数のプロセッサでのその一方の変換処理を実行した結果得られた一群の結果データを構成する複数の結果データ部分群が異なるプロセッサに割り当てられるように、上記一群の結果データを上記複数のプロセッサの間で交換するステップを有し、
上記第１から第３の変換処理は、一群の順序づけられた変換対象データに対する一群の順序づけられたフーリエ変換係数データを構成する複数のフーリエ変換係数データ部分群をそれぞれ異なるプロセッサにより生成するように定められ、
各プロセッサには、上記一群の変換対象データを構成する複数の変換対象データ部分群の一つの変換対象データ部分群がそのプロセッサに対して予め割り当てられ、
上記一群のフーリエ変換係数データのそれぞれを生成したプロセッサの順序が、上記一群の変換対象データのそれぞれが割り当てられたプロセッサの順序と同一となるように、上記交換するステップで各プロセッサに割り当てられる結果データ部分群が定められているもの。
【請求項２】各プロセッサは、そのプロセッサに対して予め割り当てられた一つの変換対象データ部分群に対して、上記第１の変換処理を他のプロセッサと並行して実行し、第１の結果データの部分群を生成し、
各プロセッサは、上記交換するステップが上記第２の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第１の変換処理の結果生成した上記一つの第１の結果データ部分群に対して、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一群の第１の部分結果データに対して、上記第２の変換処理を他のプロセッサと並行して実行して、一つの第２の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その各プロセッサで実行した上記第２の変換処理の結果生成した一つの第２の結果データ部分群に対して、上記交換するステップが上記第２の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一つの第２の結果データ部分群に対して、上記第３の変換処理を他のプロセッサと並行して実行して一つのフーリエ変換係数データ部分群を生成する請求項１記載のフーリエ変換方法。
【請求項３】上記第１、第２、第３の変換処理は、それぞれ３次元データ空間の第１、第２、第３の座標軸に関する変換処理であり、
上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数の変換対象データ部分群は、上記３次元データ空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、
上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数のフーリエ変換係数データ部分群は、上記３次元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められている請求項１記載のフーリエ変換方法。
【請求項４】複数のプロセッサを有する計算機で実行するためのフーリエ変換方法であって、
各プロセッサにより、３次元データ空間の第１の座標軸に関する第１の変換処理、上記３次元データ空間の第２の座標軸に関する第２の変換処理、上記３次元データ空間の第３の座標軸に関する第３の変換処理を順次実行し、
上記複数のプロセッサの各々による、上記第１、第２の変換処理のいずれか一方の変換処理の実行後に、上記複数のプロセッサでのその一方の変換処理を実行した結果得られた一群の結果データを構成する複数の結果データ部分群がそれぞれ異なるプロセッサに割り当てられるように、上記一群の結果データを上記複数のプロセッサの間で交換するステップを有し、
上記第１から第３の変換処理は、一群の順序づけられた変換対象データに対する一群の順序づけられたフーリエ変換係数データを構成する複数のフーリエ変換係数データ部分群をそれぞれ異なるプロセッサにより生成するように定められ、
上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数の変換対象データ部分群は、上記３次元データ空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、
上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数のフーリエ変換係数データ部分群は、上記３次

元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められ、
各プロセッサには、上記一群の変換対象データを構成する複数の変換対象データ部分群の一つの変換対象データ部分群が予め割り当てられ、
上記一群のフーリエ変換係数データのそれぞれを生成したプロセッサの順序が、上記一群の変換対象データのそれぞれが割り当てられたプロセッサの順序と同一となるように、上記交換するステップで各プロセッサに割り当てられる結果データ部分群が定められているもの。
【請求項５】各プロセッサは、そのプロセッサに対して予め割り当てられた一つの変換対象データ部分群に対して、上記第１の変換処理を他のプロセッサと並行して実行して、第１の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第２の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第１の変換処理の結果生成した第１の結果データ部分群に対して、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一つの第１の結果データ部分群に対して、上記第２の変換処理を他のプロセッサと並行して実行して、一つの第２の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第１の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第２の変換処理の結果生成した一つの第２の結果データ部分群に対して、上記交換するステップが上記第２の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一つの第２の結果データ部分群に対して、上記第３の変換処理を他のプロセッサと並行して実行して一つのフーリエ変換係数データ部分群を生成する請求項４に記載のフーリエ変換方法。
【請求項６】上記変換対象データ群が上記３次元データ空間に直方体形状に位置する格子点群に上記３次元空間に第３の座標軸、第２の座標軸、第１の座標軸の順に順次割り当てられ、
上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが、３次元係数空間に直方体形状に位置する格子点群に、当該３次元係数空間の第１、第２、第３の座標軸の順序で割り当てられるように定められている請求項４記載のフーリエ変換方法。
【請求項７】各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、
上記交換ステップが上記第１の変換処理が上記複数のプロセッサにより実行された後に実行され、
上記複数のプロセッサは、この交換ステップで、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、
各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元波数空間の第２の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、
各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む請求項４記載のフーリエ変換方法。
【請求項８】各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、
上記交換ステップが上記第２の変換処理が上記複数のプロセッサにより実行された後に実行され、
各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１、第２の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、
上記複数のプロセッサは、上記交換ステップにより、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、
各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元係数空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む請求項４記載のフーリエ変換方法。
【請求項９】複数のプロセッサを有する計算機で実行す

るためのフーリエ変換方法であって、
各プロセッサにより、３次元空間の第１、第２、第３の座標軸の座標にそれぞれ関する第１、第２、第３の変換処理を順次かつ他のプロセッサと並行して実行し、
各プロセッサが上記第１、第２の変換処理のいずれか一方を実行した後に、その一方の変換処理の結果それぞれのプロセッサで得られた複数の結果データを上記複数のプロセッサの間で交換するステップを有し、
ここで、一群の順序づけられた変換対象データが上記３次元空間に直方体の形に並べられ、
上記第１から第３の変換処理は、上記一群の変換対象データに対する一群の順序づけられた３次元空間の座標を有する複数のフーリエ変換係数データを生成するように定められ、
上記複数の変換対象データが構成する上記直方体を分割する上記３次元空間の上記第１の座標軸に垂直な複数の面の各々に含まれる複数の変換対象データが同一のプロセッサに割り当てられ、
上記交換ステップは、上記一方の変換処理の結果得られた上記複数の結果データが構成する３次元空間の直方体を、その３次元空間の第１の座標軸に垂直な複数の面で分割し直して、各面に属する複数の結果データを同一のプロセッサに割り当てるように、上記一方の変換処理の結果得られた上記複数の結果データを上記複数のプロセッサ間で交換するステップを有するもの。
【請求項１０】上記一群の順序づけられた変換対象データを上記３次元空間に直方体の形に並べられる順序は、第３の座標軸、第２の座標軸、第１の座標軸の順であり、
上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが３次元空間に第１、第２、第３の座標軸の順序で並べられるように定められている請求項９記載のフーリエ変換方法。
【請求項１１】各プロセッサがパイプライン演算器を含み、その演算器での演算の対象とするループ長がＬのときのその各プロセッサの演算性能を求めるための性能データを上記複数のプロセッサに共通に記憶し、
その性能データを用いて、上記直方体の上記第１、第２、第３の座標軸方向の長さを決定し、
その決定された上記第１、第２、第３の座標軸方向の長さを有する直方体に、上記順序づけられた複数の変換対象データを並べるステップをさらに有する請求項９記載のフーリエ変換方法。
【請求項１２】計算機により読みとり可能なプログラム記録媒体であって、複数のプロセッサを有する計算機でフーリエ変換を実行するためのプログラムを記憶し、そのプログラムは、
各プロセッサにより、第１の変換処理、第２の変換処理、第３の変換処理を順次かつ他のプロセッサと並行して実行し、
上記複数のプロセッサの各々による、上記第１、第２の変換処理のいずれか一方の変換処理の実行後に、上記複数のプロセッサでのその一方の変換処理を実行した結果得られた一群の結果データを構成する複数の結果データ部分群が異なるプロセッサに割り当てられるように、上記一群の結果データを上記複数のプロセッサの間で交換するステップを実行するようにプログラムされ、
上記第１から第３の変換処理は、一群の順序づけられた変換対象データに対する一群の順序づけられたフーリエ変換係数データを構成する複数のフーリエ変換係数データ部分群をそれぞれ異なるプロセッサにより生成するように定められ、
各プロセッサには、上記一群の変換対象データを構成する複数の変換対象データ部分群の一つの変換対象データ部分群がそのプロセッサに対して予め割り当てられ、
上記一群のフーリエ変換係数データのそれぞれを生成したプロセッサの順序が、上記一群の変換対象データのそれぞれが割り当てられたプロセッサの順序と同一となるように、上記交換するステップで各プロセッサに割り当てられる結果データ部分群が定められているもの。
【請求項１３】各プロセッサは、そのプロセッサに対して予め割り当てられた一つの変換対象データ部分群に対して、上記第１の変換処理を他のプロセッサと並行して実行し、第１の結果データの部分群を生成し、
各プロセッサは、上記交換するステップが上記第２の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第１の変換処理の結果生成した上記一つの第１の結果データ部分群に対して、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一群の第１の部分結果データに対して、上記第２の変換処理を他のプロセッサと並行して実行して、一つの第２の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その各プロセッサで実行した上記第２の変換処理の結果生成した一つの第２の結果データ部分群に対して、上記交換するステップが上記第２の変換処理の実行後に実行されるときには、その交換ステップでそのプロセッサに割り当てられた一つの第２の結果データ部分群に対して、上記第３の変換処理を他のプロセッサと並行して実行して一つのフーリエ変換係数データ部分群を生成する請求項１２記載のプログラム記録媒体。
【請求項１４】上記第１、第２、第３の変換処理は、それぞれ３次元データ空間の第１、第２、第３の座標軸に関する変換処理であり、
上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数の変換対象データ部分群は、上記３次元データ

空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、
上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数のフーリエ変換係数データ部分群は、上記３次元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められている請求項１３記載のプログラム記録媒体。
【請求項１５】計算機により読みとり可能なプログラム記録媒体であって、複数のプロセッサを有する計算機でフーリエ変換を実行するためのプログラムを記憶し、
そのプログラムは、
各プロセッサにより、３次元データ空間の第１の座標軸に関する第１の変換処理、上記３次元データ空間の第２の座標軸に関する第２の変換処理、上記３次元データ空間の第３の座標軸に関する第３の変換処理を順次実行し、
上記複数のプロセッサの各々による、上記第１、第２の変換処理のいずれか一方の変換処理の実行後に、上記複数のプロセッサでのその一方の変換処理を実行した結果得られた一群の結果データを構成する複数の結果データ部分群がそれぞれ異なるプロセッサに割り当てられるように、上記一群の結果データを上記複数のプロセッサの間で交換するステップを実行するようにプログラムされ、
上記第１から第３の変換処理は、一群の順序づけられた変換対象データに対する一群の順序づけられたフーリエ変換係数データを構成する複数のフーリエ変換係数データ部分群をそれぞれ異なるプロセッサにより生成するように定められ、
上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数の変換対象データ部分群は、上記３次元データ空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、
上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、
上記複数のフーリエ変換係数データ部分群は、上記３次元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められ、
各プロセッサには、上記一群の変換対象データを構成する複数の変換対象データ部分群の一つの変換対象データ部分群が予め割り当てられ、
上記一群のフーリエ変換係数データのそれぞれを生成したプロセッサの順序が、上記一群の変換対象データのそれぞれが割り当てられたプロセッサの順序と同一となるように、上記交換するステップで各プロセッサに割り当てられる結果データ部分群が定められているもの。
【請求項１６】各プロセッサは、上記一群の変換対象データを構成する複数の変換対象データ部分群の内のそのプロセッサに対して予め割り当てられた一つの変換対象データ部分群に対して、上記第１の変換処理を他のプロセッサと並行して実行して、第１の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第２の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第１の変換処理の結果生成した第１の結果データ部分群に対して、上記交換するステップが上記第１の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一つの第１の結果データ部分群に対して、上記第２の変換処理を他のプロセッサと並行して実行して、一つの第２の結果データ部分群を生成し、
各プロセッサは、上記交換するステップが上記第１の変換処理の実行後に実行されるときには、その各プロセッサで実行した上記第２の変換処理の結果生成した一つの第２の結果データ部分群に対して、上記交換するステップが上記第２の変換処理の実行後に実行されたときには、その交換ステップでそのプロセッサに割り当てられた一つの第２の結果データ部分群に対して、上記第３の変換処理を他のプロセッサと並行して実行して一つのフーリエ変換係数データ部分群を生成する請求項１５記載のプログラム記録媒体。
【請求項１７】上記変換対象データ群が上記３次元データ空間に直方体形状に位置する格子点群に上記３次元空間に第３の座標軸、第２の座標軸、第１の座標軸の順に順次割り当てられ、
上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが、３次元係数空間に直方体形状に位置する格子点群に、当該３次元係数空間の第１、第２、第３の座標軸の順序で割り当てられるように定められている請求項１５記載のプログラム記録媒体。
【請求項１８】各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、
上記交換ステップが上記第１の変換処理が上記複数のプ

ロセッサにより実行された後に実行され、
上記複数のプロセッサは、この交換ステップで、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、
各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元波数空間の第２の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、
各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む請求項１５記載のプログラム記録媒体。
【請求項１９】各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、
上記交換ステップが上記第２の変換処理が上記複数のプロセッサにより実行された後に実行され、
各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１、第２の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、
上記複数のプロセッサは、上記交換ステップにより、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、
各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元係数空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む請求項１５記載のプログラム記録媒体。
【請求項２０】計算機により読みとり可能なプログラム記録媒体であって、複数のプロセッサを有する計算機でフーリエ変換を実行するためのプログラムを記憶し、
そのプログラムは、
各プロセッサにより、３次元空間の第１、第２、第３の座標軸の座標にそれぞれ関する第１、第２、第３の変換処理を順次かつ他のプロセッサと並行して実行し、
各プロセッサが上記第１、第２の変換処理のいずれか一方を実行した後に、その一方の変換処理の結果それぞれのプロセッサで得られた複数の結果データを上記複数のプロセッサの間で交換するステップを実行するようにプログラムされ、
ここで、一群の順序づけられた変換対象データが上記３次元空間に直方体の形に並べられ、
上記第１から第３の変換処理は、上記一群の変換対象データに対する一群の順序づけられた３次元空間の座標を有する複数のフーリエ変換係数データを生成するように定められ、
上記複数の変換対象データが構成する上記直方体を分割する上記３次元空間の上記第１の座標軸に垂直な複数の面の各々に含まれる複数の変換対象データが同一のプロセッサに割り当てられ、
上記交換ステップは、上記一方の変換処理の結果得られた上記複数の結果データが構成する３次元空間の直方体を、その３次元空間の第１の座標軸に垂直な複数の面で分割し直して、各面に属する複数の結果データを同一のプロセッサに割り当てるように、上記一方の変換処理の結果得られた上記複数の結果データを上記複数のプロセッサ間で交換するステップを有するもの。
【請求項２１】上記一群の順序づけられた変換対象データが上記３次元空間に直方体の形に並べられる順序は、第３の座標軸、第２の座標軸、第１の座標軸の順であり、
上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが、３次元空間に、第１、第２、第３の座標軸の順序で並べられるように定められ請求項２０記載のプログラム記録媒体。
【請求項２２】各プロセッサがパイプライン演算器を含み、その演算器での演算の対象とするループ長がＬのときのその各プロセッサの演算性能を求めるための性能データを上記複数のプロセッサに共通に記憶し、
その性能データを用いて、上記直方体の上記第１、第２、第３の座標軸方向の長さを決定し、
その決定された上記第１、第２、第３の座標軸方向の長さを有する直方体に、上記順序づけられた複数の変換対象データを並べるステップをさらに有する請求項２０記載のプログラム記録媒体。
【請求項２３】上記プログラムは、各プロセッサ上で実行されるアプリケーションプログラムから呼び出され、そのアプリケーションプログラムが指定するフーリエ変換対象データに対してフーリエ変換を実行し、生成したフーリエ変換係数データをそのアプリケーションプログ

ラムに戻すライブラリである請求項１２から２２のいずれか一つに記載のプログラム記録媒体。

【請求項２４】複数のプロセッサを有する計算機で実行するためのシミュレーション方法であって、

各プロセッサにより、シミュレーションすべき物理現象を支配する方程式に基づいて、シミュレーション対象の物理空間の異なる点での少なくとも一つの物理量の値を計算し、

各プロセッサにより、その計算に当たり、算出された複数の値を表すデータにし対してフーリエ変換を実行するステップとを有し、

上記フーリエ変換を実行するステップは、請求項１から１１のいずれか一つにより実行されるもの。

【請求項２５】上記物理現象は物理的な装置の動作に関連する物理現象であり、

上記算出された複数の値に基づいて、上記物理的な装置を設計するためのデータを生成するステップをさらに有する請求項２４記載のシミュレーション方法。

【請求項２６】上記物理的な装置は、半導体装置である請求項２５記載のシミュレーション方法。

【請求項２７】上記物理現象は気象であり、

上記算出された複数の値に基づいて、気象予測として報じるためのデータを生成するステップをさらに有する請求項２４記載のシミュレーション方法。

【請求項２８】計算機により読みとり可能なプログラム記録媒体であって、複数のプロセッサを有する計算機でシミュレーションを実行するためのプログラムを記憶し、そのプログラムは、請求項２４から２７のいずれか一つによりシミュレーションを実行するようにプログラムされているもの。

【発明の詳細な説明】

【発明の属する技術分野】本発明は、複数のプロセッサを有する計算機で実行するのに適したフーリエ変換方法に係り、とくに、ベクトル演算器を内蔵する複数のプロセッサからなるベクトル並列計算機で実行するのに適したフーリエ変換方法に関する。

【従来の技術】科学技術計算において頻繁に利用される処理の一つに、フーリエ変換がある。フーリエ変換は、物理現象のシミュレーションその他に使用される。フーリエ変換は、ある実数区間で定義された複素数値をとる関数ｆ（ｘ）を複素指数関数ｅｘｐ（ｉｋｘ）の重ね合わせとして表す処理であり、計算機上で実現する場合には、扱いうる点の数が有限であることから、複素数の点列f_0, f_1, ．．．, f_{N-1}をＮ個の複素指数関数ｅｘｐ（２πｉｋｊ／Ｎ）（ただし、ｋ＝０，１，．．．，Ｎ－１で、ｉは虚数単位、πは円周率）の重ね合わせとして表す処理となる。すなわち、f_0, f_1, ．．．, f_{N-1}が与えられたときに、式１ａにより重ね合わせの係数c_0, c_1, ．．．, c_{N-1}を求めるのがフーリエ変換である。各点f_jの値は、これらの係数を用いると、式１ｂによりあらわされる。

【数１】

$$c_k = (1/N)\sum_{j=0}^{N-1} f_j \exp(-2\pi ikj/N) \quad (k=0,1,\ldots,N-1) \qquad (1a)$$

$$f_j = \sum_{k=0}^{N-1} c_k \exp(2\pi ikj/N) \quad (j=0,1,\ldots,N-1) \qquad (1b)$$

しかし、この定義に基づいて計算を行うと、式の数がＮ本あり、各式がＮ個の項から成るため、複素指数関数ｅｘｐ（－２πｉｋｊ／Ｎ）の計算に加えて、複素数の加算と乗算が約N^2回必要である。そこで実際には、アルゴリズム上の工夫により計算量を約ＮｌｏｇＮのオーダーに減少させた高速フーリエ変換という手法が広く使われている。高速フーリエ変換を並列計算機上で効率的に行うための手法として、従来、転置アルゴリズムとバイナリ・エクスチェンジと呼ばれる２つの手法が提案されている（たとえばV. Kumar, A. Grama, A. Gupta and G. Karypis: "Introductionto Parallel Computing, The Benjamin/Cummings Publishing Company, 1994参照）。前者はプロセッサ間の通信を計算途中の一箇所にまとめて行う方式、後者はプロセッサ間で通信を行いながら計算を進める方式であり、プロセッサの台数をｐとすると、通信の回数は前者がｐ－１、後者が$\log_2 p$で、通信１回あたりに送るデータ量は、前者がＮ／p^2、後者がＮ／２ｐである。後者は前者に比べて通信の回数が少なくて済むため、通信のセットアップ時間が支配的となる小規模問題では通信時間が少なくて済むという利点があるが、通信すべきデータの総量は多くなるため、大規模データの場合には前者が有利となる。半導体デバイスの特性計算、電子状態計算、気象予測のための計算などの科学技術計算では、数万から数百万に上る変数を扱う大規模シミュレーションが必要である。このような大規模問題を扱う手段としては、並列計算機が有力である。並列計算機は数十個から数万個に上る多数の高速プロセッサをネットワークで結んだシステムであり、従来の逐次型計算機に比べ、プロセッサ台数を増やすことでピーク性能をいくらでも高めることができるという利点を持つ。さらに、最近の並列計算機では、各プロセッサで、一連のデータに対して同じ演算を高速に実行するできるように演算器が構成されていることも多い。とくに、各プロセッサに、そのような演算器として同じ演算を複数のデータに対してパイプライン的に実行するベクトル演算器を有するベクトル並列計算機も開発されている。ベクトル並列計算機の中には、このベクトル演算器による演算を指定するベクトル命令を実行できるものもある。さらに、メモリとベクトル演算器の間に複数のベクトルレジスタが設けられている並列計算機も

ある。これらのベクトルレジスタはメモリと演算器のデータの転送時間が処理時間に及ぼす時間を軽減している。より高速にシミュレーションを実行可能になっている。また、厳密にはベクトル並列計算機ではないが、ベクトル並列計算機に類似の並列計算機として、ある演算を実行する演算器がベクトル演算器でなくても、一連のデータに対してその演算を高速に実行できるように構成されている演算器を使用する並列計算機も多い。フーリエ変換は科学技術計算でもっともよく使われる処理の一つであり、最近では並列計算機用のライブラリとして提供されることも多い。たとえば日立製作所編「プログラムプロダクトＨＩ－ＵＸ／ＭＰＰ行列計算副プログラムライブラリＭＡＴＲＩＸ／ＭＰＰ」参照。並列計算機で実行する大規模のシミュレーションがフーリエ変換を実行する場合には前述の転置アルゴリズムが使用されることが多い。上に記載したように、ベクトル型並列計算機あるいはそれらに類似の並列計算機で転置アルゴリズムを実行する場合には、変換すべき一次元空間の点列データを３次元空間に直方体状に並べ、これに対してたとえばＹ方向の変換、Ｘ方向の変換、Ｚ方向の変換を順次行うことによって、全データ点列に対して高速フーリエ変換を行ったのと同一の結果を得る。より具体的には、フーリエ変換の対象となる一次元のデータf_0, f_1, ．．．．, f_{N-1}を入力し、各辺の長さがＮＸ，ＮＹ，ＮＺの直方体状に並べる。ここで、ＮＸ，ＮＹ，ＮＺはＮＸ＊ＮＹ＊ＮＺ＝Ｎを満たす整数である。データを直方体状に並べるに当たっては、原点からたとえばＺ方向にデータを並べていき、ＮＺ個のデータを並べ終わったら次はＸ座標を１だけ増やしてデータを並べ、これを繰り返してＮＸ＊ＮＺ個のデータを並べ終わったら次はＹ座標を１だけ増やしてデータを並べる、という操作を行う。このようにデータを並べた後、直方体をＺ軸に垂直にスライスし、こうしてできる各面を並列計算機の一つのプロセッサに割り当てる。次に、Ｙ方向の変換を行う。プロセッサへの入力データf_jの割り当て方式より、各ＸＹ平面は１台のプロセッサに担当されているから、この変換処理は通信なしに各プロセッサで独立に行える。次に、同様にして各プロセッサで独立にＹ方向の変換の結果データに対してＸ方向の変換を行う。Ｘ方向の変換の終了後、プロセッサ間でのＸ方向の変換の結果データの入れ替えを行い、今度はその結果データが構成する直方体をＸ軸に垂直にスライスし、こうしてできる各面を一つのプロセッサに割り当てる。この処理を転置と呼び、各プロセッサが自分以外の全プロセッサとデータの交換を行う必要がある。転置の終了後、今度は各プロセッサで独立にＺ方向の変換を行う。以上で、直方体状に並べられた一次元入力データf_jのフーリエ変換が終了し、直方体状に並べられた、重ね合わせの係数を表す出力データc_kが求まる。出力データc_kの並び方は、原点からまずＹ方向に、Ｙ方向にＮＹ個行ったら次はＸ座標が１だけ増え、ＸＹ平面上にＮＸ＊ＮＹ個のデータが並んだら次はＺ座標が１だけ増えるという順で並ぶ。上記の転置アルゴリズムでは、入力データf_jの分割は、f_jを第ＭＯＤ（ｊ，ｐ）番のプロセッサが担当するという形でデータがプロセッサ間で分割されている。このデータ分割形式はサイクリック分割と呼ばれる。データ分割形式はデータのプロセッサへの割り当ての順序を表すものでもあり、本明細書ではデータ分割形式のことを割り当て順序あるいは割り当て態様とも呼ぶことがある。一方、出力データc_kの分割は、ＮＹ個の連続するデータを１台のプロセッサが担当するブロックサイクリック分割となり、入力データf_jとはプロセッサ間のデータ分割形式が異なる。上記の転置アルゴリズムでは、入力データf_jの並べ方およびそのデータのプロセッサへの分割の仕方より分かるように、入力データの分割形式は、f_jを第ＭＯＤ（ｊ，ｐ）番のプロセッサが担当するというサイクリック分割となる。一方、転置後のデータのプロセッサへの分割の仕方、および変換で得られた出力データc_kの並び方より分かるように、出力データの分割形式は、ＮＹ個の連続するデータを１台のプロセッサが担当するというブロックサイクリック分割となる。しかし多くの応用では、フーリエ変換と逆フーリエ変換とを対にして用い、しかも逆フーリエ変換はフーリエ変換プログラムを流用して行うため、フーリエ変換の入力データと出力データが同じデータ分割形式（データ割り当て順序）になっている方が都合がよい。そのため、従来の高速フーリエ変換方法では、以上の処理に従ってブロックサイクリック分割の出力データc_kを得た後、再びプロセッサ間でデータの転送を行い、データc_kをサイクリック分割に直して出力する必要がある。

【発明が解決しようとする課題】本発明者の検討の結果、以上の従来のフーリエ変換方法では、フーリエ変換係数の計算後に行うデータ分割形式（データ割り当て順序）の変更のためのプロセッサ間でのデータ転送が、フーリエ変換時間の短縮を妨げていることが分かった。したがって、本発明の目的は、フーリエ変換係数の計算後にデータ分割形式の変更のためにデータ転送を行わなくても、フーリエ変換結果データがフーリエ変換対象データと同一のデータ分割形式（データ割り当て順序）を持ち得るフーリエ変換方法を提供することである。

【課題を解決するための手段】上記目的を達成するため、本発明によるフーリエ変換方法は、各プロセッサにより、第１の変換処理、第２の変換処理、第３の変換処理を順次実行し、上記複数のプロセッサの各々による、上記第１、第２の変換処理のいずれか一方の変換処理の実行後に、上記複数のプロセッサでのその一方の変換処理を実行した結果得られた一群の結果データを構成する複数の結果データ部分群が異なるプロセッサに割り当てられるように、上記一群の結果データを上記複数のプロセッサの間で交換するステップを有する。上記第１から

第３の変換処理は、一群の順序づけられた変換対象データに対する一群の順序づけられたフーリエ変換係数データを構成する複数のフーリエ変換係数データ部分群をそれぞれ異なるプロセッサにより生成するように定められ、各プロセッサには、上記一群の変換対象データを構成する複数の変換対象データ部分群の一つの変換対象データ部分群がそのプロセッサに対して予め割り当てられ、上記一群のフーリエ変換係数データのそれぞれを生成したプロセッサの順序が、上記一群の変換対象データのそれぞれが割り当てられたプロセッサの順序と同一となるように、上記交換するステップで各プロセッサに割り当てられる結果データ部分群が定められているもの。より具体的には、上記第１、第２、第３の変換処理は、それぞれ３次元データ空間の第１、第２、第３の座標軸に関する変換処理であり、上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、上記複数の変換対象データ部分群は、上記３次元データ空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、上記複数のフーリエ変換係数データ部分群は、上記３次元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められている。本発明の具体的な態様によるフーリエ変換方法では、上記変換対象データ群の各々は、上記３次元データ空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、上記複数の変換対象データ部分群は、上記３次元データ空間の第３の座標軸の座標値が同じであり、上記３次元データ空間の第１、第２の座標軸の座標値が異なる全ての変換対象データが同一の変換対象データ部分群に含まれるように定められ、上記フーリエ変換係数データ群の各々は、３次元係数空間の直方体形状に位置する格子点群の一つの座標をそれぞれ有し、上記複数のフーリエ変換係数データ部分群は、上記３次元係数空間の第１の座標軸の座標値が同じであり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる全てのフーリエ変換係数データが同一のフーリエ変換係数データ部分群に含まれるように定められる。更に具体的には、上記変換対象データ群が上記３次元データ空間に直方体形状に位置する格子点群に上記３次元空間に第３の座標軸、第２の座標軸、第１の座標軸の順に順次割り当てられ、上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが、３次元係数空間に直方体形状に位置する格子点群に、当該３次元係数空間の第１、第２、第３の座標軸の順序で割り当てられるように定められている。更に具体的な態様では、各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、上記交換ステップが上記第１の変換処理が上記複数のプロセッサにより実行された後に実行され、上記複数のプロセッサは、この交換ステップで、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元波数空間の第２の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元波数空間の第２、第３の座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む。更に具体的な他の態様では、各プロセッサが上記第１の変換処理により生成する上記一つの第１の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元データ空間の第２の座標軸の座標値と上記３次元係数空間の第１の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含み、上記交換ステップが上記第２の変換処理が上記複数のプロセッサにより実行された後に実行され、各プロセッサが上記第２の変換処理により生成する上記一つの第２の結果データ部分群は、上記３次元データ空間の第３の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１、第２の座標軸の座標値が異なる値を有する全ての複数の第２の結果データを含み、上記複数のプロセッサは、上記交換ステップにより、上記３次元係数空間の第１の座標軸の座標値が所定の同じ値であり、上記３次元係数空間の第１の座標軸の座標値と上記３次元データ空間の第３の座標軸の座標値が異なる値を有する全ての複数の第１の結果データを含む第１の結果データ部分群が同一のプロセッサに割り当てられるように、上記複数のプロセッサが生成した一群の第１の結果データを上記複数のプロセッサの間で交換し、各プロセッサが上記第３の変換処理により生成する上記一つのフーリエ変換係数部分群は、上記３次元係数空間の第１の座標軸の座標値が所定の値であり、上記３次元係数空間の第２、第３の

座標軸の座標値が異なる値を有する全ての複数のフーリエ変換係数を含む。本発明のより具体的な態様では、各プロセッサにより、３次元空間の第１、第２、第３の座標軸の座標にそれぞれ関する第１、第２、第３の変換処理を順次かつ他のプロセッサと並行して実行し、各プロセッサが上記第１、第２の変換処理のいずれか一方を実行した後に、その一方の変換処理の結果それぞれのプロセッサで得られた複数の結果データを上記複数のプロセッサの間で交換するステップを有する。ここで、一群の順序づけられた変換対象データが上記３次元空間に直方体の形に並べられ、上記第１から第３の変換処理は、上記一群の変換対象データに対する一群の順序づけられた３次元空間の座標を有する複数のフーリエ変換係数データを生成するように定められ、上記複数の変換対象データが構成する上記直方体を分割する上記３次元空間の上記第１の座標軸に垂直な複数の面の各々に含まれる複数の変換対象データが同一のプロセッサに割り当てられ、上記交換ステップは、上記一方の変換処理の結果得られた上記複数の結果データが構成する３次元空間の直方体を、その３次元空間の第１の座標軸に垂直な複数の面で分割し直して、各面に属する複数の結果データを同一のプロセッサに割り当てるように、上記一方の変換処理の結果得られた上記複数の結果データを上記複数のプロセッサ間で交換するステップを有する。とくに、望ましくは、上記一群の順序づけられた変換対象データを上記３次元空間に直方体の形に並べられる順序は、第３の座標軸、第２の座標軸、第１の座標軸の順であり、上記第１から第３の変換処理は、上記複数のフーリエ変換係数データが３次元空間に第１、第２、第３の座標軸の順序で並べられるように定められている。さらに望ましくは、本発明によるフーリエ変換方法は、各プロセッサがパイプライン演算器を含み、その演算器での演算の対象とするループ長がＬのときのその各プロセッサの演算性能を求めるための性能データを上記複数のプロセッサに共通に記憶し、その性能データを用いて、上記直方体の上記第１、第２、第３の座標軸方向の長さを決定し、その決定された上記第１、第２、第３の座標軸方向の長さを有する直方体に、上記順序づけられた複数の変換対象データを並べるステップをさらに有する。本発明によるプログラム記憶媒体は、上記いろいろのフーリエ変換方法のいずれかを実行するようにプログラムされたプログラムが記憶する。さらに、本発明によるシミュレーション方法は、上記いろいろのフーリエ変換方法のいずれかを使用してシミュレーションを実行する。本発明による他のプログラム記憶媒体は、上記シミュレーション方法を実行するようにプログラムされたプログラムを記憶する。

【発明の実施の形態】以下、本発明に係るフーリエ変換方法、それを用いるシミュレーション方法およびプログラムを図面に示したいくつかの実施の形態を参照してさらに詳細に説明する。なお、以下においては、同じ参照番号は同じものもしくは類似のものを表わすものとする。また、第２の実施の形態以降では、第１の実施の形態との相違点を主に説明するに止める。

＜発明の実施の形態１＞

（１）装置の概略構成

本発明によるフーリエ変換方法を実行するための並列計算機システムの一例を図１に示す。並列計算機２８は、それぞれがメモリ２６を備えた複数のプロセッサ２７と、プログラムおよびデータを格納するための複数の外部記憶装置３１から構成され、これらの装置は、内部データ転送ネットワーク２９を介して相互にデータを交換可能なように構成されている。外部記憶装置３１には、たとえば、多くのユーザの利用に供するために並列計算機２８に予め準備された複数のプログラムライブラリ４４とそれらが使用するデータ３０等が記憶される。各プロセッサのメモリ２６は、いわゆるローカルメモリであり、このメモリに記憶されたデータに割り当てられるアドレスは、そのプロセッサで定められたローカルなアドレス空間に属するアドレスであり、この種のメモリは一般に分散メモリと呼ばれ、この種のメモリを有する計算機は分散メモリ型の並列計算機と呼ばれる。並列計算機２８は、各プロセッサ２９が、一連のデータ要素からなるベクトルデータに対して同じ演算をパイプライン的に連続して実行できるベクトル演算器（図示せず）を備えるベクトル並列計算機であると仮定する。これらのプロセッサ２７内の特定の一つのプロセッサには、ユーザが操作可能な計算機、たとえばワークステーション１がＬＡＮ等のネットワーク２を介して接続されている。この計算機は他の計算機たとえばパーソナルコンピュータでもよい。このワークステーション１には、並列計算機２８に対する指示あるいはデータを入力するための入力装置３（典型的には、キーボードとマウス）と、並列計算機２８からの計算結果を出力するための出力装置２９（典型的には、表示装置と印刷装置）が接続されている。なお、ワークステーション１内には、並列計算機２８に送るべきプログラムおよびそのプログラムで使用するデータを記憶する記憶装置（図示せず）も設けられている。上記特定のプロセッサは、並列計算機２８内で計算を司るプロセッサの役目とユーザ用のワークステーション１との通信の役目とを兼ねる。すなわち、このプロセッサは、ワークステーション１から送付されるプログラムとデータを受信し、それらを外部記憶装置３１の一つに記憶し、その後、並列計算機２８の内部に記憶された適当なプログラムにより、ユーザ指定のプログラムを複数のプロセッサ（具体的には全プロセッサ）にロードし、ユーザ指定のデータの異なる部分を、それぞれそれらのプロセッサの異なるものに割り当て、そのユーザ指定のプログラムを起動する。しかしながら、本発明によるフーリエ変換方法を実施するためには、並列計算機２８は、複数のプロセッサを有することが必要であるが、

それ以外の点では特に限定した構造を有しなくてもよいことは言うまでもない。並列計算機２８は、ベクトル並列計算機であると仮定したが、このベクトル演算器はごく一部の演算のみを実行でき、他の演算はベクトル演算器ではないスカラ演算器で実行されてもよい。さらに、並列計算機２８は、対してこのような演算器を有しなくてもよい。もちろん、一連のデータに対する同じ演算を高速に実行できるように構成されている演算器を有することが望ましい。また、並列計算機２８は、メモリ２９と演算器（図示せず）の間に複数のベクトルレジスタを有しないと仮定するが、これらのレジスタが使用することはより望ましいことである。さらに、それらのプロセッサの具体的な構造あるいはそれらの間のデータ転送ネットワークの構造、あるいはそれらのプロセッサと入力装置あるいは出力装置との接続形態がいろいろであっても、本発明はそれらの並列計算機に適用可能である。たとえば、ワークステーションと通信可能な複数のプロセッサが設けられていてもよく、また、ワークステーションと通信可能な少なくとも一つのプロセッサが計算用のプロセッサとは別に設けられていてもよい。また、実行すべきプログラムとデータを並列計算機２８に送付する方法は他の方法に依ってもよいことは明らかである。ユーザは上記複数のプロセッサ２７を使用して種々の計算を実行できる。最も典型的な計算は、物理現象などのシミュレーションであり、たとえば、地球の気象の予測もシミュレーションにより行われる。半導体デバイスの設計も、半導体デバイスの物理的な動作をシミュレーションして行われている。このようなシミュレーションを並列計算機を使用して実行する場合、シミュレーション対象の物理領域を複数の部分領域に区分し、各部分領域を一つのプロセッサに割り当て、そのプロセッサにおいて、その部分領域についてのシミュレーションを、たとえば一つまたは複数の物理量に関する偏微分方程式を解いて実行することが多い。この場合、シミュレーションに使用される複数のプロセッサは、同じシミュレーションプログラムを互いに並列に実行する。したがって、このようなプログラムは並列プログラムとも呼ばれる。各プロセッサが実行するシミュレーションプログラムが使用するデータは異なる。たとえばシミュレーション領域の位置と形状を表すデータ、シミュレーションすべき物理量の初期値、シミュレーション領域の物質に関する物質定数、あるいは各部分領域に関する境界条件など異なる。各プロセッサは、計算の途中で得られた結果データを他の適当なプロセッサに転送し、あるいは他のプロセッサから計算結果データを受け取り、さらにシミュレーションを続けていく。このシミュレーションプログラムの中にはフーリエ変換を使用するものもある。本実施の形態では、いろいろのシミュレーションの利用に供するために、本発明によるフーリエ変換方法にしたがってフーリエ変換を実行するようにプログラムされたフーリエ変換ライブラリがいずれかの外部記憶装置３１に記憶される。さらにプロセッサ間の通信を実行するための通信ライブラリも外部記憶装置３１に記憶される。シミュレーションプログラムは、上記フーリエ変換ライブラリあるいは通信ライブラリを必要な時点でコールするようにプログラムされる。並列計算機２８は、ワークステーション１から送信されたユーザ指定のシミュレーションプログラムと、そのシミュレーションプログラムが使用するライブラリ（今の場合には上記フーリエ変換ライブラリと上記通信ライブラリ）を各プロセッサにロードする。さらに、並列計算機２８は、それぞれのプロセッサでシミュレーションプログラムが使用する、ワークステーション１から送信されたユーザ指定のデータをそれぞれのプロセッサにロードする。なお、シミュレーションプログラムは全プロセッサにロードされてもよく、一部のプロセッサにロードされてもよいが、以下では簡単化のために、シミュレーションプログラムは、全てのプロセッサにロードされると仮定する。上記ライブラリあるいは上記シミュレーションプログラムは、並列計算機２８の命令セットあるいはハード構造の特徴、ソフトウエア上の制約等を反映するコンパイラによりコンパイルされたものである。本発明によりフーリエ変換方法を実行する上記ライブラリあるいは上記シミュレーションプログラムに上記ライブラリに含まれたフーリエ変換のためのプログラム部分を組み込んだプログラムを磁気記憶装置のようなプログラム記録媒体に記憶して販売できる。

（２）並列高速フーリエ変換の原理

すでに述べたごとく、フーリエ変換は、Ｎ個の入力データf_0, f_1, ．．．, f_{N-1}からＮ個の出力データc_0, c_1, ．．．, c_{N-1}を、式１ａを用いて計算する処理である。入力データf_j、出力データc_kは、実数データであっても複素データであってもよい。入力データf_j、出力データc_kはそれぞれ実空間のデータ、波数空間のデータと呼ばれることがある。すなわち、入力データf_jの添え字jは、一次元の実空間の格子点の座標を表し、出力データc_kの添え字ｋは、一次元の波数空間の格子点の座標を表すと考えることができる。言い換えると、上記の式によるフーリエ変換は、一次元の実空間のデータを一次元の波数空間のデータに変換する処理である。したがって、本明細書では、入力データf_jの添え字jを一次元実空間の格子点座標あるいは単に座標と呼び、出力データc_kの添え字ｋを一次元波数空間の格子点の座標あるいは単に座標と呼ぶことがある。あるいは、それらのデータはその座標を有すると呼ぶことがある。しかし、入力データf_j、出力データc_kが実際にはそのような実空間、波数空間に属するデータでなくてもよい。一般に、並列計算機を使用して演算を行う場合、できるだけ多くのプロセッサが互いに並列に動作する時間を増大するとともに、プロセッサ間のデータ通信の総回数を少なくすることが望ましいことが知られている。データ通

信は、プロセッサ内部の計算時間に比べて時間が掛かる上に、通信はあるプロセッサからのデータの送信と他のプロセッサでの受信となからなり、受信側のプロセッサでは、ある処理を実行する前に他のプロセッサからそこでの演算結果データを受信するようにプログラムされた場合、そのプロセッサは、受信すべき演算結果データが受信されるまで、その処理を開始することができない。したがって、各プロセッサでは、通信の発生に伴い、受信待ち時間が増大し、他のプロセッサと並列に動作する時間が減少する。したがって、並列計算機で演算を高速に行うには、プロセッサ間の通信の総回数を減らすことが望ましいことが知られている。このことは演算としてフーリエ変換を並列計算機で実行する場合も同じであ

$$j = j_y*NX*NZ + j_x*NZ + j_z$$

（ただし、 $j_x=0,1,\ldots,NX-1$, $j_y=0,1,\ldots,NY-1$, $j_z=0,1,\ldots,NZ-1$）

【数３】

$$k = k_z*NX*NY + k_x*NY + k_y$$

（ただし、 $k_x=0,1,\ldots,NX-1$, $k_y=0,1,\ldots,NY-1$, $k_z=0,1,\ldots,NZ-1$）

ここで、記号＊は乗算を表す。この置換は、１次元実空間の格子点座標ｊ、１次元の波数空間の格子点座標ｋを、それぞれ３次元の実空間の格子点座標（j_x, j_y, j_z）、３次元の波数空間の格子点座標（k_x, k_y, k_z）に写像することであるとも言える。３次元の実空間の座標（j_x, j_y, j_z）は、１次元の実空間の座標ｊから次式により計算される。

$j_z=MOD(j/NZ, NX)$

$j_y=(j/(NX*NZ))\downarrow$

$j_z=MOD(j, NZ)$

ここで、（　　）↓は、括弧内の数値の整数部分のみを表し、小数点以下を切り捨てることを表す。したがって、ｊが０，１，２，３，，，（Ｎ－１）と変化したときに、（j_x, j_y, j_z）は、（０，０，０），（０，０，１），（０，０，２），（０，０，３），，，（０，０，ＮＺ－１）と変化し、さらに、（１，０，０），（１，０，１），（１，０，２），（１，０，３），，，（１，０，ＮＺ－１）と変化し、この変化を（ＮＸ－１，０，ＮＺ－１）まで変化した後に、j_yを１に変えて上記変化を座標（ＮＸ－１，ＮＹ－１，ＮＺ－１）に達するまで繰り返す。すなわち、一次元の順次異なる座標点ｊに対応する３次元の座標点（j_x, j_y, j_z）は、Ｚ方向、Ｘ方向、Ｙ方向の順に変化する。本明細書では、このようにフーリエ変換の対象となる一次元実空間のデータ f_0, f_1, ．，f_{N-1}を３次元実空間のデータに写像することを、簡単化のために各辺の長さがＮＸ，ＮＹ，ＮＺの直方体状に並べるともいう。以上の置換は、言い換えると、図２に示すように、原点から

る。このためには、演算で使用するデータをどのプロセッサに割り当てるか、いつプロセッサ間で演算結果データを交換するかが重要な問題である。並列計算機でフーリエ変換を行うには、従来の転置アルゴリズムでは、変換対象データ f_j を以下のようにして３次元の実空間のデータに写像し、それを用いて変換対象データを割り当てるプロセッサを決定することになる。いま、ＮＸ，ＮＹ，ＮＺをＮＸ＊ＮＹ＊ＮＺ＝Ｎを満たす正の整数とし、１次元の添字ｊ，ｋを３次元の添字（j_x, j_y, j_z）、（k_x, k_y, k_z）に次の式２、３によって置換する。

【数２】

．．．（２）

．．．（３）

まずＺ方向にデータを並べていき、ＮＺ個のデータを並べ終わったら次はＸ座標を１だけ増やしてデータを並べ、これを繰り返してＮＸ＊ＮＺ個のデータを並べ終わったら次はＹ座標を１だけ増やしてデータを並べるという操作を行うことと等価である。但し、図２では、Ｎは５１２であり、ＮＸ，ＮＹ，ＮＺはともに８に等しいと仮定した。３次元の波数空間の座標（k_x, k_y, k_z）は、１次元の波数空間の座標ｋから次式により計算される。

$k_x=MOD(k/NY, NX)$

$k_y=MOD(k, NY)$

$k_z=(k/(NX*NY))\downarrow$

したがって、ｋが０，１，２，３，，，（ＮＸ＊ＮＹ＊ＮＺ－１）と変化したときに、（k_x, k_y, k_z）は、（０，０，０），（０，１，０），（０，２，０），（０，３，０），，，（０，ＮＹ－１，０）と変化し、さらに、（１，０，０），（１，１，０），（１，２，０），（１，３，０），，，（１，ＮＹ－１，０）と変化し、この変化を（ＮＸ－１，ＮＹ－１，０）まで変化した後に、k_zを１に変えて上記変化を座標（ＮＸ－１，ＮＹ－１，ＮＺ－１）に達するまで繰り返す。すなわち、一次元の順次異なる座標点ｋに対応する３次元の座標点（k_x, k_y, k_z）は、Ｙ方向、Ｘ方向、Ｚ方向の順に変化する。したがって、求めるべきフーリエ変換係数 c_k とそれに対応する３次元の波数空間の座標（k_x, k_y, k_z）との関係は図３に示すとおりになる。但し、図３では、Ｎは５１２であり、ＮＸ，ＮＹ，ＮＺはともに８に等しいと仮定した。なお，転置アルゴ

リズム自体は、１次元空間のデータ列f_j、それに対するフーリエ変換結果データc_kを２次元空間のデータ$f_{jx,jy,ckx,ky}$に変換して行うこともできる。この場合には、１次元のフーリエ変換を２次元のフーリエ変換に置き直すことになる。しかし、ここで記載するように、１次元空間のデータ列f_j、それに対するフーリエ変換結果データc_kを３次元空間のデータ$f_{jx,jy,jz}$と$c_{kx,ky,kz}$に変換してフーリエ変換を行うのは，並列計算機の個々のプロセッサがベクトル演算器を持つ場合に、そのベクトル演算器をうまく利用するためである。この場合には、１次元のフーリエ変換を３次元のフーリエ変換に置き直すことになる。すなわち，以下の実施の形態でも述べるように，変換の各ステップでは，Ｘ方向、Ｙ方向、Ｚ方向のうちのどれかの方向について変換を行い，残りの２方向のうちの１方向を用いて並列化を行い，更に残りの１方向を用いてベクトル化を行う。このため，データを３つの方向を持つ直方体状に並べる必要がある。原理的には，式２，３と同様の変換を行って，データを２次元空間あるいは４次元以上の空間に並べ直すこともできるが，２次元では並列化とベクトル化の両方を行うには次元が足りず，また，４次元以上では不要な次元ができ，その分だけベクトル化対象のループ長が短くなってしまうので性能的に不利である。そのため，ベクトル演算器を持つ並列計算機上で高速フーリエ変換を行うには，データを式２，３のように３次元に並べ直すことがことが望ましい。このようなデータの変換は、各プロセッサがベクトル演算器を持たない場合にも演算の高速化に有効な場合が多い。置換式２，３を使用すると、式１ａは次のように書き換えられる。

【数４】

$$
\begin{aligned}
c_k &= c_{kx,ky,kz} \\
&= (1/N)\Sigma_{jz=0}^{Nz-1}\Sigma_{jx=0}^{Nx-1}\Sigma_{jy=0}^{Ny-1} f_{jx,jy,jz} \\
&\quad *\exp(-2\pi i(k_z*NX*NY+k_x*NY+k_y) \\
&\quad (j_y*NX*NZ+j_x*NZ+j_z)/(NX*NY*NZ)) \\
&= (1/N)\Sigma_{jz=0}^{Nz-1}\exp(-2\pi i((k_z*NX*NY+k_x*NY+k_y)j_z) \\
&\quad /(NX*NY*NZ)) \\
&\quad *(\Sigma_{jx=0}^{Nx-1}\exp(-2\pi i((k_x*NY+k_y)j_x)/(NX*NY)) \\
&\quad *(\Sigma_{jy=0}^{Ny-1} f_{jx,jy,jz}*\exp(-2\pi i k_y j_y/NY)))
\end{aligned}
$$

（ただし、$k_x=0,1,\ldots,NX-1$，
$k_y=0,1,\ldots,NY-1$，
$k_z=0,1,\ldots,NZ-1$）　　．．．（４）

さらに、この変換式は次の３式で表される３ステップの変換をそれらの式の順に順次実行することにより実現される変換であることが分かる。

【数５】

$$c'_{jx,ky,jz}=\Sigma_{jy=0}^{Ny-1} f_{jx,jy,jz}*\exp(-2\pi i k_y j_y/NY) \quad (5)$$

【数６】

$$
\begin{aligned}
c''_{kx,ky,jz} &= \Sigma_{jx=0}^{Nx-1} c'_{jx,ky,jz} \\
&\quad *\exp(-2\pi i(k_x+k_y/NY) j_x/NX) \qquad ．．．(6)
\end{aligned}
$$

【数７】

$$
\begin{aligned}
c_{kx,ky,kz} &= \Sigma_{jz=0}^{Nz-1} c''_{kx,ky,jz} \\
&\quad *\exp(-2\pi i(k_z+k_x/NY+k_y/(NX*NY)) j_z/NZ) \quad ．．(7)
\end{aligned}
$$

式５は、j_x、j_zが特定の値であり、j_yの値が異なるＮＹ個の入力データ$f_{jx,jy,jz}$に対するフーリエ変換を、j_x、j_zが採りうる値の組合わせの数（ＮＸ＊ＮＺ組）だけ行い、それにより上記二つの実空間座標（j_x、j_z）の組のひとつにそれぞれ対応する複数（ＮＸ＊ＮＺ）組の、３次元の波数空間の一つの座標（k_y）に関する一次変換結果データ（$c'_{jx,ky,jz}$）（k_y＝０～ＮＹ－１、j_x＝０～ＮＸ－１、j_z＝０～ＮＺ－１）を得る処理を表す。式６も、複素指数関数の中にk_y／ＮＹという余分な項が入る以外はフーリエ変換と同じ変換を表し、具体的には、この式は、j_z、k_yが特定の値であり、j_xの値が異なるＮＸ個の一次変換結果データ$c'_{jx,ky,jz}$に対してフーリエ変換と類似の変換を、j_z、k_yが採りうる値の組合わせの数（ＮＹ＊ＮＺ組）だけ行い、それにより、座標j_zの異なる値に対する、３次元の波数空間の二つの座標（k_x、k_y）に関する２次変換結果データ（$c''_{kx,ky,jz}$）（k_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１、j_z＝０～ＮＺ－１）を得る処理を表す。式７も、複素指数関数の中にk_x／ＮＹ＋k_y／（ＮＸ＊ＮＹ）という余分な項が入る以外はフーリエ変換と同じ変換を表し、具体的には、この式は、k_x、k_yが特定の値であり、j_zの値が異なるＮＺ個のデータ$c''_{kx,ky,jz}$に対してフーリエ変換と類似の変換を、k_x、k_yが採りうる値の組合わせの数（ＮＸ＊ＮＹ組）だけ行い、それにより３次元の波数空間の３つの座標（k_x, k_y, k_z）に関する最終的なフーリエ変換結果データ（$c_{kx,ky,kz}$）（k_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１、k_z＝０～ＮＺ－１）を得る処理を表す。したがって、これらの３つの変換は、すべて高速フーリエ変換のアルゴリズムを用いて実行することができ

る。以下では、これらの変換をそれぞれＹ方向の変換、Ｘ方向の変換、Ｚ方向の変換と呼ぶ。本明細書では、これらの変換を簡単化のためにそれぞれＹ方向のフーリエ変換、Ｘ方向のフーリエ変換、Ｚ方向のフーリエ変換と呼ぶこともある。ここで、Ｘ方向等は、式２、３で定めた座標変換できまる方向である。すなわち、一次元の順次異なる座標点ｊに対応して最初に順次変化する座標がＺ座標であり、その後に変化する座標がＸ座標であり、最後に変化する座標がＹ座標である。座標変換式を式２、３から変更すれば、Ｘ方向等の変換の内容が変わるのは言うまでもない。したがって、本明細書では、より一般的には、これらの変換は以下の変換を指す。Ｙ方向の変換とは、式５により例示されたように、実空間の第１、第３の座標軸の座標（j_x，j_z）が特定の値であり、第２の座標軸の座標（j_y）の値が異なる複数（ＮＹ）個の入力データ$f_{jx,jy,jz}$に対してフーリエ変換を行い、３次元実空間の第１、第３の座標軸の座標（j_x，j_z）の組のひとつにそれぞれ対応する複数（ＮＸ＊ＮＺ）組の、３次元波数空間の第２の座標軸の座標（k_y）に関連する一次変換結果データ（$c'_{jx,ky,jz}$）（j_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１、j_z＝０～ＮＺ－１）を得る処理を指す。あるいは、言い換えると、Ｙ方向の変換は、入力データ$f_{jx,jy,jz}$に対して、実空間の第２の座標軸に関してフーリエ変換を行い、３次元波数空間の第２の座標軸の座標と、３次元実空間の第１の座標軸の座標と、第３の座標軸の座標との関数である一次変換結果データを得る処理を指すとも言える。さらに、Ｘ方向の変換とは、式６により例示されたように、上記第３の実空間座標系の第１の座標系の座標（j_z）と３次元波数空間の第２の座標系の座標（k_y）とが特定の値であり、上記第１の実空間座標系の座標（j_x）の値が異なる複数（ＮＸ）個の一次変換結果データ（$c'_{jx,ky,jz}$）に対してフーリエ変換に類似の変換を行い、上記第３の実空間座標軸の座標（j_z）の異なる値の一つにそれぞれ対応する複数（N_z）個の、上記３次元の波数空間の第１、第２の座標軸の座標（k_x、k_y）に関連する２次変換結果データ（$c''_{kx,ky,jz}$）（k_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１、j_z＝０～ＮＺ－１）を得る処理を指す。あるいは、言い換えると、Ｘ方向の変換は、一次変換結果データに対して、３次元実空間の第１の座標軸に関してフーリエ変換に類似の変換を行い、３次元実空間の第３の座標軸の座標と、３次元波数空間の第１、第２の座標軸の座標との関数である２次変換結果データを得る処理を指すとも言える。さらに、Ｚ方向の変換とは、式７により例示されたように、３次元波数空間の第１、第２の座標系の座標（k_x，k_y）とが特定の値であり、３次元実空間の第３の座標系の座標（j_z）の値が異なる複数（ＮＺ）個の２次変換結果データ（$c''_{kx,ky,jz}$）に対してフーリエ変換に類似な変換を行い、３次元波数空間の第１、第２、第３の座標軸の座標（k_x，k_y，k_z）に関連する、入力データに対する最終的なフーリエ変換結果データ（$c_{kx,ky,kz}$）（k_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１、k_z＝０～ＮＺ－１）を得る処理を指すとも言える。あるいは、言い換えると、Ｚ方向の変換は、２次変換結果データに対して、３次元実空間の第３の座標軸に関してフーリエ変換に類似の変換を行い、３次元波数空間の第１、第２、第３の座標軸の座標の関数である最終的なフーリエ変換結果データを得る処理を指すとも言える。Ｙ方向の変換は、式５にて示されるように、ＮＸ＊ＮＺ組のデータに対する互いに独立な変換からなる。同様に、Ｘ方向の変換は、式６にて示されるように、ＮＹ＊ＮＺ組のデータに対する互いに独立な変換からなる。同様に、Ｚ方向の変換は、式７にて示されるように、ＮＸ＊ＮＹ組のデータに対する互いに独立な変換からなる。従来の転置アルゴリズムによるフーリエ変換方法では、この特徴を利用してプロセッサ間の通信を少なくするように、変換対象データを並列計算機の異なるプロセッサに割り当てている。すなわち、式２に従って、また、図２に例示されるように、変換対象データf_j（ｊ＝０～Ｎ）を直方体状に並べ、３次元実空間のＺ軸に並行な平面でこのデータを分割し、図５に例示するように、j_z＝０，１，，，７をそれぞれ有する複数のデータはプロセッサ０，１，，７に割り当てられている。すなわち、特定の値のＺ座標j_zを有する全ての変換対象データは、それらのＸ座標j_x、Ｙ座標j_yの値に依らないで同一のプロセッサに割り当てる。図２では、Ｎ＝５１２，ＮＸ＝ＮＹ＝ＮＺ＝８と仮定し、図４ではプロセッサの総数ＮＰＵ＝８と仮定したが、これらの数値がここに仮定の数値と異なる場合でも、特定の値のＺ座標j_zを有する全ての変換対象データは、それらのＸ座標、Ｙ座標の値に依らないで同一のプロセッサに割り当てればよい。たとえば、プロセッサの総数ＮＰＵ＝ＮＸ＝ＮＺとし、ＮＹ＝（Ｎ／（ＮＸ＊ＮＺ））↑とすればよい。ここで、（　）↑は、括弧内の数値の小数点以下を切り上げた後の整数を示す。たとえば、Ｎ＝５１２、ＮＰＵ＝４のときには、ＮＸ＝ＮＺ＝４、ＮＹ＝３２であればよい。このようなデータの割り当てを行った後、フーリエ変換を以下のように実行する。この方法では式５によるＹ方向の変換と式６によるＸ方向の変換とは、プロセッサ間の通信を使用しないで行うことができる。

（ステップ１）Ｙ方向の変換

まず、Ｙ方向の変換を実行する。式５から分かるように、Ｙ方向の変換では、Ｘ座標j_xとＺ座標j_zが特定の値であり、Ｙ座標j_yが異なる複数の変換対象データに対してフーリエ変換が実行される。しかし、これらの複数のデータは同じプロセッサに割り当てられている。こうして、各プロセッサでは、式５にしたがって、プロセッサ間の通信を使用しないで、Ｘ座標j_xとＺ座標j_zが特定の値であり、波数空間のＹ座標k_yがいろいろの値を有

する複数の一次変換結果データが得られる。

（ステップ２）Ｘ方向の変換

次に、Ｘ方向の変換を実行する。式６から分かるように、Ｘ方向の変換では、Ｚ座標j_zと波数空間のＹ座標k_yが特定の値であり、Ｘ座標j_xが異なる複数の一次変換結果データに対してフーリエ変換に類似の変換が実行される。しかし、これらの複数のデータは同じプロセッサでのＹ方向の変換によりすでに得られている。こうして、各プロセッサは、式６にしたがって、他のプロセッサとの通信をしないで、そのプロセッサに割り当てられた、３次元実空間のＺ座標j_zの特定の値に対する、３次元波数空間の座標（k_x，k_y）に関連するＮＸ＊ＮＹ個の２次変換結果データ（$c''_{kx,ky,jz}$）（k_x＝０～ＮＸ－１、k_y＝０～ＮＹ－１）を得る。

（ステップ３）データの転置

式７によるＺ方向の変換を行うには、３次元波数空間の座標（k_x，k_y）の特定の値と、３次元実空間のＺ座標j_zの全ての値に対して得られた２次変換結果データ（$c''_{kx,ky,jz}$）が必要である。そこで、各プロセッサに、３次元波数空間の座標k_xの特定の値を割り当てて、Ｚ方向の変換を実行するに必要なデータをプロセッサ間で転送する処理が実行される。すなわち、各プロセッサへの座標k_xの値の割り当てでは、プロセッサ０，１，２，，，７には、k_x＝０，１，２，３，，，を順次割り当てる。このことは、図５に示すように、３次元波数空間を、そのk_x軸に垂直な平面で分割して、分割後の部分空間の各々を一つのプロセッサに割り当てることを意味する。各プロセッサが、そのプロセッサに割り当てられた３次元波数空間の座標k_xの特定の値と、３次元波数空間の座標k_yの全ての値と、３次元実空間のＺ座標j_zの全ての値とに対して得られた全ての２次変換結果データ（$c''_{kx,ky,jz}$）を使用してＺ方向の変換を実行できるように、全プロセッサ間で２次変換結果データの転送が行われる。このデータ転送は、データの転置あるいはデータの並び替えとも言われる。すなわち、各プロセッサは、３次元実空間のＺ座標j_zの全ての値と、３次元波数空間の座標（k_x，k_y）の全ての値との組に対して得られた２次変換結果データ（$c''_{kx,ky,jz}$）（但し、k_x＝特定値、k_y＝０～ＮＹ－１、k_z＝０～（ＮＺ－１））の内、自プロセッサが生成しなかったデータを他のプロセッサから受信するように、全プロセッサの間で２次変換結果データを転送する。

（ステップ４）Ｚ方向の変換

各プロセッサは、そのプロセッサに割り当てられた３次元波数空間の座標k_xの特定の値と、３次元波数空間の他の二つの座標k_y，k_zの全ての値を有する最終的なフーリエ変換結果データ$c_{kx,ky,kz}$（但し、k_x＝特定値、k_y＝０～ＮＹ－１、k_z＝０～ＮＺ－１）を計算する。各プロセッサはこの計算を互いに並列に実行できる。

（ステップ５）データの転置

しかし以上の処理だけでは、変換対象データf_jと変換結果データc_kのデータ分割形式が異なり、実用上不便であるという問題がある。すなわち、変換対象データf_jは、図２に示すように、３次元実空間のデータに写像され、後者のデータは、図４にしたがって分割されて複数のプロセッサに割り当てられた。したがって、図６に示すように、変換対象データf_jの分割は、f_jを第ＭＯＤ（j，p）番のプロセッサが担当するサイクリック分割となる。一方、変換結果データc_kは、図３に示すように、３次元波数空間のデータに写像され、３次元波数空間は図５にしたがって分割されて複数のプロセッサに割り当てられた。したがって、図７に示すように、変換結果データc_kのデータ分割は、ＮＹ個の連続するデータを１台のプロセッサが担当するブロックサイクリック分割となる。多くの応用では、変換対象データをフーリエ変換して得られる変換結果データに対してある処理を施し、その処理結果に対して再び逆フーリエ変換を行う。逆フーリエ変換はフーリエ変換のプログラムを流用して行われることが多い。すなわち、次の逆フーリエ変換の式

【数８】$f_j=\Sigma_{k=0}^{N-1} c_k \exp(2\pi ikj/N)$

（ただし、j＝0,1,...,N-1）　　．．．（８）

を変形すると次式が得られる。

【数９】$f_j=(\Sigma_{k=0}^{N-1} c_k^* \exp(-2\pi ikj/N))^*$

（ただし、j＝0,1,...,N-1）　　．．．（９）

ここで、＊印は複素共役を示す。したがって、式１ａと９の比較より明らかなように、逆フーリエ変換は、フーリエ変換結果データの複素共役をフーリエ変換し、得られた結果データの複素共役を取ることに等価である。したがって、原理的には、逆フーリエ変換はフーリエ変換のプログラムを流用して実行できることが分かる。しかし、並列計算機でフーリエ変換を実行するときには、変換対象データと変換結果データとのデータ分割形式が異なると、いずれかのプロセッサに割り当てられた変換対象データに対する変換結果データがそのプロセッサに割り当てられていないことになり、そのプロセッサは、その変換結果データを他のプロセッサから受信しないとフーリエ変換のプログラムを流用して逆フーリエ変換を行うことができなくなる。このような不便を避けるため、従来の転置アルゴリズムを使ったフーリエ変換プログラムでは、上記Ｚ方向の変換を実行した後に、３次元フーリエ変換結果データ$c_{kx,ky,kz}$のプロセッサへの割り当てを変更し、再びプロセッサ間でフーリエ変換結果データ$c_{kx,ky,kz}$の転置（入れ替え）を行い、フーリエ変換結果データ$c_{kx,ky,kz}$のデータ分割をサイクリック分割に直すのが一般的であった。すなわち、図８に示すように、３次元波数空間を、Ｙ座標軸に垂直な平面で切断し、同じＹ座標値k_yを有するフーリエ変換結果データ$c_{kx,ky,kz}$を同一のプロセッサに割り当てる。具体的に

は、k_y＝０，１，２，，，を有するフーリエ変換結果データ$c_{kx,ky,kz}$を順次プロセッサ０，１，２，，，に割り当てる。各プロセッサが、この割り当てにしたがってそのプロセッサに割り当てられたフーリエ変換結果データ$c_{kx,ky,kz}$の内、自プロセッサが生成しなかったデータを他のプロセッサから受信するように、全プロセッサの間でフーリエ変換結果データ$c_{kx,ky,kz}$を転送する。こうして、サイクリック分割された３次元フーリエ変換結果データが得られる。こうして、得られた最終的３次元フーリエ変換結果データ$c_{kx,ky,kz}$から目的とする１次元フーリエ変換結果データc_kは式３より得ることができる。データc_kとその三次元座標k_x，k_y，k_zとの関係は図３に示されたとおりである。しかし、本発明者による検討によれば、従来のデータのデータ入れ替えのためのプロセッサ間での余分な通信は、並列化効率を低下し、フーリエ変換に必要な処理時間を増大する原因であることが判明した。そこで本発明では、従来の転置アルゴリズムにおけるデータの分割方式を見直し、プロセッサ間のデータ転送量の削減の側面から最適なデータ分割方式を以下のようにして決定した。従来の転置アルゴリズムは、Ｙ方向、Ｘ方向、Ｚ方向の各変換と、プロセッサ間でのデータの入れ替えを行う転置操作から構成される。そのアルゴリズムでは、Ｙ方向の変換を行う際に、変換対象データの空間をＺ軸に垂直な複数の平面で切り、各面を一つのプロセッサに割り当てて、次にＸ方向の変換を行う際には、その割り当てをそのまま使用し、さらにＺ方向の変換を行う際には、Ｘ軸に垂直な複数の平面で変換対象データの空間を切り、各面を一つのプロセッサに割り当てていた。これにより、その変換そのものは、プロセッサ間のデータ転送なしに実行できた。しかし、たとえば最初のＹ方向の変換を行う際には、変換の対象となる同一のＸ座標とＺ座標を持つＮＹ個のデータが１台のプロセッサ上にありさえすれば、その変換そのものは、プロセッサ間のデータ転送なしに実行できる。したがって、この変換対象データの分割は、Ｚ軸に垂直な平面によってではなく、Ｘ軸に垂直な平面によって行ってもよい。このことは、Ｘ方向、Ｚ方向の変換についても言える。したがって、望ましいデータ分割方式が満たすべき第一の条件は、「ある方向の変換を行うときには、その変換の変換対象データをその方向以外の方向に垂直な複数の平面で分割してプロセッサへのデータ割り当てをする」というデータ分割形式が、Ｙ方向、Ｘ方向、Ｚ方向のすべてに採用されていることである。今ひとつ考慮すべきことは転置のためのデータ転送回数である。たとえば、Ｙ方向、Ｘ方向、Ｚ方向の変換を行うとき、変換対象データをそれぞれＺ軸、Ｙ軸、Ｘ軸に垂直な複数の平面で切って分割するというデータ分割方式は、上記の第１の条件を満たすが、このデータ分割方式では、データ分割形式がＸ方向、Ｚ方向の変換を行うときという２回にわたって変更され、その変更の度に転置のためのデータ転送が必要になる。したがって、望ましいデータ分割方式が満たすべき条件として、上記の第１の条件に加えて、「データ分割形式の変更のための転置処理は一回に限る」という第二の条件を付加する。これら２つの条件を満たすデータ分割方式を数え上げた結果を図９に示す。上記第１、第２の条件を満たすデータ分割方式は４通りあり、これらの内で、フーリエ変換対象データのデータ分割形式とフーリエ変換結果データのデータ分割形式が同一のデータ分割方式が求めるものである。方式１が従来の転置アルゴリズムで採用されているものである。方式４は入力データがＸ方向に分割、出力データがＹ方向に分割だから、図３および図６と照らし合わせてみると、フーリエ変換対象データがブロックサイクリック分割、フーリエ変換結果データがサイクリック分割であり、方式１とはデータ分割形式がちょうど逆になってはいるものの、これらの二つの種類のデータの間でデータ分割形式が異なるという方式１と同様の欠点を抱えていることがわかる。一方、方式２では、従来の転置アルゴリズムと同じく、フーリエ変換対象データがＺ方向に沿って分割され、Ｙ方向の変換、Ｘ方向の変換も従来の転置アルゴリズムと同じように実行されるが、Ｚ方向の変換は、従来の転置アルゴリズムと異なり、Ｘ方向の変換の結果データがＹ方向に沿って分割された後に実行される。Ｘ方向の変換とＺ方向の変換の間では、データ転置が必要である。図２および図３と照らし合わせてみると、フーリエ変換対象データもフーリエ変換結果データもサイクリック分割になることが分かる。したがって、方式２のデータ分割方式を採用すると、フーリエ変換係数の計算後にプロセッサ間でデータ転送を行わなくても、フーリエ変換対象データとフーリエ変換結果データとが同じデータ分割形式を保つ。この方式２では、フーリエ変換は具体的には以下のようにして実行される。以下に記載するＹ方向、Ｘ方向、Ｚ方向の変換はＦＦＴのアルゴリズムにより計算される。

（ステップａ）Ｙ方向の変換

Ｙ方向の変換は、従来の転置アルゴリズムについて既に述べたステップ１の要領で実行される。既に述べたごとく、フーリエ変換対象データのデータ分割は、サイクリック分割である。

（ステップｂ）Ｘ方向の変換

さらに、Ｘ方向の変換は、Ｙ方向の変換の結果データに対して、従来の転置アルゴリズムについて既に述べたステップ２の要領でなされる。

（ステップｃ）データ転置

式７によるＺ方向の変換を行うには、３次元波数空間の座標（k_x，k_y）の特定の値と、３次元実空間のＺ座標j_zの全ての値に対して得られた２次変換結果データ（$c''_{kx,ky,jz}$）が必要である。方式２では、従来の転置アルゴリズムと異なり、Ｘ方向の変換で得られた２次変換結果データ（$\grave{c}''_{kx,ky,jz}$）は、Ｙ軸に垂直な複数

の平面で分割される。このことは、図８に示すように、３次元波数空間を、そのk_y軸に垂直な複数の平面で分割して、分割後の部分空間（上記複数の平面）の各々を一つのプロセッサに割り当てることを意味する。すなわち、各プロセッサへ座標k_yの特定の値を割り当てる。具体的には、k_x＝０，１，２，３，，，の２次変換結果データ（$c''_{kx,ky,jz}$）を順次プロセッサ０，１，２，，，７に割り当てる。この割り当てに従い、Ｚ方向の変換を実行するに必要なデータをプロセッサ間で転送する処理が実行される。各プロセッサが、そのプロセッサに割り当てられた３次元波数空間の座標k_yの特定の値と、３次元波数空間の座標k_xの全ての値と、３次元実空間のＺ座標j_zの全ての値とに対して得られた全ての２次変換結果データ（$c''_{kx,ky,jz}$）を使用してＺ方向の変換を実行できるように、全プロセッサ間で２次変換結果データの転送が行われる。すなわち、各プロセッサは、３次元実空間のＺ座標j_zの全ての値と、３次元波数空間の座標k_xの全ての値と、座標k_yの特定の値との組に対して得られた２次変換結果データ（$c''_{kx,ky,jz}$）（但し、k_x＝０～（ＮＸ－１）、k_y＝特定値、k_z＝０～（ＮＺ－１））の内、自プロセッサが生成しなかったデータを他のプロセッサから受信するように、全プロセッサの間で２次変換結果データを転送する。

（ステップｄ）Ｚ方向の変換

各プロセッサは、そのプロセッサに割り当てられた３次元波数空間の座標k_yの特定の値と、３次元波数空間の他の二つの座標k_x、k_zの全ての値を有する最終的なフーリエ変換結果データ$c_{kx,ky,kz}$（但し、k_x＝０～ＮＸ－１、k_y＝特定値、k_z＝０～ＮＺ－１）を計算する。各プロセッサはこの計算を互いに並列に実行できる。この結果、座標k_y＝０，１，２，３，，，に対応するフーリエ変換係数c_0，c_1，c_2，，，が、順次プロセッサ０，１，２，，，で生成され、全フーリエ変換結果データ$c_{kx,ky,kz}$は、プロセッサ間でサイクリックに分割されていることが分かる。本実施の形態では、上記データ分割方式２を使用する。なお、データ分割方式３も後に詳細に説明するように、フーリエ変換対象データもフーリエ変換結果データもサイクリック分割になっている。したがって、この方式３も使用することができる。後に述べるように、実際にフーリエ変換ライブラリを構成する場合に、方式３は、方式２に比べて、ライブラリが生成する複素指数関数の値のテーブルのサイズが小さくてよいという利点を有する。なお、計算機により入力データf_jに対して以上の変換を実施するときには、一般に配列データが使用される。すなわち、同じプロセッサでＹ方向の変換を施すべき一群のデータは、３次元配列に格納され、その配列に対してＹ方向の変換が実行される。その結果得られた１次変換結果データは、同じ配列あるいは他の３次元配列に格納されてもよい。他の方向の変換も先に実行された変換の結果データを格納する配列に対してなされる。また、プロセッサ間でのデータの転置も各プロセッサが生成した配列の内容を交換するようになされる。したがって、このようにそれらの変換において同じ３次元配列を使用するときには、その３次元配列の各次元のインデックスは、あるときには３次元実空間の各座標軸に対応し、他の時にはある方向の変換後の結果データが属する３次元空間の各座標軸に対応し、最終的にはフーリエ変換係数が属する３次元波数空間の各座標軸に対応することになる。しかし、このように同じ配列を異なる種類の一群のデータの格納に使用された場合でも、ある時点でその配列に格納されている一群のデータは、その一群のデータが属する３次元空間に属し、その配列の各インデックスは、その一群のデータが属する３次元空間のいずれかの座標軸を表すことには変わりはない。したがって、本発明を実施するにあたって一群のデータを格納するのに使用する配列の具体的な構造は特定のもの限定されない。さらに、以上の原理で説明したいくつかの３次元空間のいずれか一つに属する一群のデータを格納する配列の構造が、その３次元空間に直接対応しないものであっても、その配列に含まれた各データは、その３次元空間の座標を有すると見なすことができ、以上の原理説明がその配列に対してもあてはまるのは言うまでもない。

(3) 多次元フーリエ変換への応用

以上、１次元フーリエ変換のためのアルゴリズムを説明したが、本方式は多次元フーリエ変換の場合へも簡単に拡張できる。次式の２次元フーリエ変換を例に採って説明する。

【数１０】

$$c_{k1,k2}=(1/N_1N_2)\Sigma_{j1=0}^{N1-1}\Sigma_{j2=0}^{N2-1}f_{j1,j2}\\ *\exp(-2\pi i(k_1j_1/N_1+k_2j_2/N_2))$$

（ただし、k_1=0,1,...,N_1-1, k_2=0,1,...,N_2-1）．．．（１０）

この式は、次式１１に変形できる。この式１１は、更に　　くことができる。次の２ステップからなる変換式１１ａ、１１ｂとして書

【数１１】

$$c_{k1,k2}=(1/N_2)\Sigma_{j2=0}^{N2-1}\exp(-2\pi ik_2j_2/N_2)\\ *(1/N_1)\Sigma_{j1=0}^{N1-1}f_{j1,j2}\\ *\exp(-2\pi ik_1j_1/N_1)\qquad\text{．．．（１１）}$$

$$c'_{k1,j2}=(1/N_1)\Sigma_{j1=0}^{N1-1}f_{j1,j2}\\ *\exp(-2\pi ik_1j_1/N_1)\qquad\text{．．（１１ａ）}$$

$$c_{k1,k2}=(1/N_2)\Sigma_{j2=0}^{N2-1} c'_{k1,j2} *\exp(-2\pi i k_2 j_2/N_2)$$

. . . （１１ｂ）

したがって、２次元フーリエ変換は、まず式１１ａのようにＮ$_1$個のデータに対する１次元フーリエ変換をＮ$_2$組行い、次に式１１ｂのようにＮ$_2$個のデータに対する１次元フーリエ変換をＮ$_1$組行うことに帰着する。したがって、これらの１次元フーリエ変換において、本発明の方式を適用できる。本発明の方式を用いて並列計算機上で２次元フーリエ変換を行うには、２次元データ$f_{j1,j2}$に対し、添え字j_1の方向（以下、これを第１方向と呼ぶ）にサイクリック分割を行う。すなわち、第i番目のプロセッサに$f_{m*NPU+i,j2}$（但し、m=0,1,...,((N1/NPU)-1)、j2=0,1,...,N2)番目の要素を割り当てる。ここで、ＮＰＵはプロセッサの台数である。すると、式１１ａのステップは、j_2が同じＮ$_1$個の要素の間での１次元フーリエ変換をＮ$_2$組行うことであり、このＮ$_1$個の要素はプロセッサ間にサイクリック分割されているから、このステップは本発明の方式による１次元フーリエ変換をＮ$_2$組行うことに帰着する。変換後のデータ$c'_{k1,j2}$は、第１方向にサイクリック分割されている。次に式１１ｂのステップでは、j_1が同じＮ$_2$個の要素の間での１次元フーリエ変換をＮ$_1$組行うが、これらＮ$_2$個の要素は同一プロセッサ上にあるため、この変換は通信なしに各プロセッサごとに独立に行える。以上により２次元フーリエ変換が完了し、変換後のデータ$c_{k1,k2}$は第１方向にサイクリック分割される。なお、以上では２次元の場合を示したが、より次元の大きい場合も、第１方向にサイクリック分割を行い、第１方向の変換のみを本発明の方式を用いて行い、以下の変換はプロセッサごとに独立に行うことにより、本発明のフーリエ変換方式を適用可能である。

（４）並列高速フーリエ変換ライブラリ

図１に戻り、並列計算機２８上で使用される高速フーリエ変換ライブラリは、具体的には、たとえば以下のように構成される。但し、本発明を適用したフーリエ変換ライブラリは、これに限定されないことは言うまでもない。本ライブラリは、全てのプロセッサにロードされ、そのプロセッサ内のシミュレーションプログラムから必要に応じてサブルーチンとしてコールされる。サブルーチン名称をＦＦＴ１Ｄとし、これを実行するには

CALL FFT1D (NX, NY, NZ, NPU, F, TB, IOPT, IER)

のように所定の引数を指定して、いずれかのプロセッサにロードされたすべてのシミュレーションプログラムから同時にコールする必要がある。ここで、Ｎ＝ＮＸ＊ＮＹ＊ＮＺはフーリエ変換対象データf_jの個数、ＮＰＵはプロセッサ台数、Ｆはライブラリのコール時はフーリエ変換対象データf_j、ライブラリからのリターン時はフーリエ変換結果データc_kを格納する配列、ＴＢは複素指数関数の値を格納するテーブル、ＩＯＰＴはサブルーチンの機能を指定する入力、ＩＥＲは実行時エラーが生じたか否かを示す出力である。ここで、配列Ｆは各プロセッサがそれぞれ持つ部分配列である。フーリエ変換の原理説明で説明したように、全入力データ（フーリエ変換対象データ）f_jは、図２のように３次元実空間に直方体状に配置され、各プロセッサには、この直方体の内、一つまたは複数の特定のＺ座標を有するＺ軸に垂直な平面に属する入力データが割り当てられる。この割り当てられた入力データが、上記引数Ｆで指定される部分配列に格納されている。すなわち、フーリエ変換対象データとフーリエ変換結果データは、ともにサイクリック分割されるので、第ｉ番目のプロセッサは、ｍ＊ＮＰＵ＋ｉ（ｍ＝０，１，．．．，Ｎ／ＮＰＵ－１）番目の要素のみを持つ。すなわち、第ｉ番目のプロセッサの配列Ｆには、Ｎ個の入力データ列f_j（ｊ＝０～Ｎ－１）の内、次式で示されるように、一群の入力データ$f_{m*NPU+i}$を格納する。

Ｆ（ｍ）＝$f_{m*NPU+i}$(m=0,1,...,N/NPU- 1)

したがって各プロセッサの持つ配列Ｆの大きさはＮ／ＮＰＵである。また、ＴＢは、第１回目のコールで計算した複素指数関数の値を格納しておくテーブルであり、２回目のコールからはここに格納した値を再利用することにより、新たな計算が不要となる。また、第１回目のコールではＩＯＰＴ＝１を指定し、このときは複素指数関数のテーブルを作成する。ＩＯＰＴ＝２は２回目以降のコールを意味し、このときは既にＴＢに格納されている値を用いる。本ライブラリのフローチャートを図１０に示す。本ライブラリは、コールされると（ステップ４５）、まず引数をチェックする（ステップ４６）。すなわち、Ｎ＝ＮＸ＊ＮＹ＊ＮＺとＮＰＵとが１以上の整数であるかどうか、ＩＯＰＴが１または２の値であるかどうかなど、引数の有効性を調べる。入力データに無効な値が入っていた場合は、ＩＥＲ＝１０００と設定して（処理４７）リターンする。次に、他のプロセッサに本ライブラリがコールされたことを通知する（ステップ４８）。この通知は、実際には、そのライブラリがロードされている通信ライブラリに、他の全てのプロセッサに当該プロセッサでのライブラリコールの発生を通知することを要求し、その通信ライブラリが、その発生を他の全てのプロセッサに通知するメッセージを送信し、それぞれの他のプロセッサでは、そこにロードされた通信ライブラリが、このメッセージを受信して、そのプロセッサでロードされた本ライブラリに、送信元のプロセッサでの本ライブラリのコールを通知する。次に、ライブラリが引数で指定した通りにＮＰＵ台のプロセッサでコールされているかどうかをチェックする（ステップ４９）。このチェックは、上に述べた他の全てのプロセッサから本ライブラリに対するライブラリコールが発生したとの通知を受信したか否かに基づいて行われる。この

条件が満たされていない場合は、ＩＥＲ＝２０００と設定して（ステップ５０）リターンする。次にＩＯＰＴの値をチェックし（ステップ５１）、ＩＯＰＴ＝１の場合は、現在のコールが、最初のコールである。したがって、そのコール元のプロセッサでのフーリエ変換を実行するための準備を行う。具体的には、Ｘ、Ｙ、Ｚの方向の変換のために、そのプロセッサで式５，６，７で使用する複素指数関数の値を前もって計算し、複素指数関数のテーブルを生成し、配列ＴＢとして格納する（ステップ５５）。計算すべき複素指数関数の値は、そのプロセッサに対するデータの割り当てにより定まる。すなわち、Ｘ、Ｙ、Ｚの方向の変換の各々においてそのプロセッサが処理すべきデータの３次元実空間の座標j_x，j_y，j_zと３次元実空間の座標k_x，k_y，k_zとを決定し、この結果により、式５から７の複素指数の偏角が採りうるいろいろの値を決定し、それぞれの偏角に対する余弦関数の値と正弦関数の値を計算し、配列ＴＢに格納する。上記決定では、各方向での変換に使用されるデータ分割形式とそのコール元のプロセッサに予め割り当てられたプロセッサ番号と、式２、３が使用される。このプロセッサ番号は、シミュレーションプログラムのロード時に予め各プロセッサに並列計算機２８により指定されるものである。各方向での変換に使用されるデータ分割形式は、使用されるデータ分割方式、本実施の形態では前述の方式２、により定まる。なお、ＩＯＰＴ＝１でない場合は、現在のコールが、２回目以降のコールである。このようなコールは、ライブラリのコール元のシミュレーションプログラムが、異なる物理量に対するフーリエ変換を行うようにプログラムされている場合において生じる。たとえば、シミュレーションプログラムが、第１の物理量に対するフーリエ変換のために本ライブラリをコールした後に、第２の物理量に対するフーリエ変換のために本ライブラリを再度コールした場合である。この場合、第２の物理量を表すフーリエ変換対象データも第１の物理量を表すフーリエ変換対象データと同じ添え字を有することが多い。この場合には、第２の物理量に対するフーリエ変換の実行時に、先に配列ＢＴに格納した複素指数の値が使用できる。したがって、ＩＯＰＴ＝１でない場合は、ステップ５５を実行しない。次に、Ｙ方向の変換を行う（ステップ５６）。本実施の形態では、全プロセッサが持つフーリエ変換対象データを仮想的に図２のような各辺の長さが引数ＮＸ，ＮＹ，ＮＺの直方体状に並べ、図４に示すように、特定の座標を有するフーリエ変換対象データを同一のプロセッサに割り当てられる。このＹ方向の変換では、既にステップ１あるいはステップａとして述べたように、各プロセッサは、同じＸ座標とＺ座標とを持つＮＹ個のデータについて、高速フーリエ変換が式５に従い行う。このようなデータの組は全部でＮＸ＊ＮＺ組あるため、結局、ＮＸ＊ＮＺ個の独立なＮＹ次の高速フーリエ変換を行うことになる。プロセッサへのデータの割り当て方式より、各ＸＹ平面は１台のプロセッサに担当されているから、この変換処理は通信なしに各プロセッサで独立に行える。本ライブラリの場合、本ライブラリにより各プロセッサが処理すべき変換対象データは、そのプロセッサで実行されるシミュレーションプログラムにより、引数Ｆで指定される配列として、そのプロセッサのメモリ（２６（図１））にコール前に格納されている。その配列Ｆの添え字と３次元実空間の座標j_x，j_y，j_zとの関係は、データ分割形式により定まる。したがって、このＹ方向の変換では、この関係を使用して配列Ｆ内の変換対象データに対して式５で指定される変換を実行する。変換で得られた一次変換結果データはそのプロセッサのメモリに記憶される。具体的には、各プロセッサでは、本ライブラリがコールされると、適当なタイミングで（たとえば、ステップ４６で入力データに無効な値が入っていないと判定された時）、各プロセッサは、データ格納用の３次元配列及び第１、第２、第３の３次元の作業配列をメモリ上に確保する。ここでは、データ格納用の３次元配列は、３次元のインデックスの長さが引数ＮＸ、ＮＹ、ＮＺに等しい。以下ではこれらの３次元の作業配列もデータ格納用の３次元配列と同じ大きさを有すると仮定する。しかし、これらの３次元の作業配列は、以下に説明するデータを格納できる大きさを有すればよく、したがって、これらの作業配列の大きさは適宜変更可能である。さらに、これらの３次元の作業配列の構造も、その使用目的に合致する限り、変更することができる。データ格納用の３次元配列には、上記引数が指定する配列Ｆに含まれるデータ点列を以下のようにして格納できる。そのプロセッサに、図２のＺ軸に垂直な一つの平面が割り当てられているときには、その一つの平面に属するＮＸ＊ＮＹ個の入力データが、それぞれのデータのＸ、Ｙ、Ｚ座標に対応する、上記データ用３次元配列のインデックスを有する位置に格納される。そのプロセッサにＺ軸に垂直な複数の平面が割り当てられたときには、各面のデータは同様にして、上記データ用３次元配列の対応するインデックスの位置に格納される。各プロセッサでは、Ｙ方向の変換はこのデータ格納用の配列を使用して、Ｚ座標が特定の値を有し、Ｙ座標とＸ座標がいろいろの値を有する一群の入力データに対して、式５により行なわれる。このとき、Ｚ座標が特定の値を有し、Ｘ座標が異なる一群の入力データに対してＹ方向の変換が高速フーリエ変換アルゴリズム（ＦＦＴ）を用いて実行される。この変換の実行にあっては、Ｘ座標が異なる一群のデータに対して、プロセッサ内のベクトル演算器（図示せず）が使用され、パイプライン的に計算が実行される。その結果得られる１次変換結果データ$c'_{jx,ky,jz}$（但し、j_x＝０～ＮＸ－１，k_y＝０～ＮＹ－１，j_z＝特定値）は、第１の３次元の作業配列の、これらの座標値j_x，k_y，j_zに対応するインデックスの

ところに格納される。したがって、図２の場合、各プロセッサでは、図２の一つの平面上の一群の入力データに対する１次変換結果データ$c'_{jx,ky,jz}$が、第１の３次元作業配列の、特定の座標j_zを有する平面上に格納される。もし、図２において、Ｚ軸に垂直な複数の平面に属する入力データがそのプロセッサに割り当てられているときには、それぞれのＺ面に対応する、上記第１の３次元作業配列内の、Ｚ軸に垂直な複数の平面のそれぞれに対応する１次変換結果データ$c'_{jx,ky,jz}$が格納される。Ｙ方向の変換の終了後、同様にしてＸ方向の変換を行う（ステップ５７）。すなわち、各プロセッサは、すでにステップ２あるいはステップｂとして述べたように、各プロセッサは、Ｙ方向の変換で得られた一次変換結果データに対して式６で指定される変換を実行する。変換で得られた２次変換結果データはそのプロセッサのメモリに記憶される。この変換処理も通信なしに各プロセッサで独立に行える。具体的には、各プロセッサでは、Ｘ方向の変換は、Ｚ座標が特定の値を有し、Ｘ座標とk_y座標とがいろいろの値を有する一群の１次変換結果データ$c'_{jx,ky,jz}$に対して、式６により行なわれる。このとき、Ｚ座標が特定の値を有し、k_y座標が異なる一群の入力データに対してＸ方向の変換が高速フーリエ変換アルゴリズム（ＦＦＴ）を用いて実行される。この変換は、上記第１の３次元作業配列を使用して実行される。この変換の実行にあっては、k_y座標が異なる一群のデータに対して、プロセッサ内のベクトル演算器が使用され、計算がパイプライン的に実行される。その結果得られる２次変換結果データ$c''_{kx,ky,jz}$（但し、k_x＝０～ＮＸ－１，k_y＝０～ＮＹ－１，j_z＝特定値）は、第２の３次元作業配列の、これらの座標値k_x，k_y，j_zに対応するインデックスのところに格納される。したがって、図２の場合、各プロセッサでは、２次変換結果データ$c''_{kx,ky,jz}$は、第２の３次元作業配列の、特定の座標j_zを有する一つの平面に格納される。もし、図２において、Ｚ軸に垂直な複数の平面に属する入力データがそのプロセッサに割り当てられているときには、それぞれのＺ面に対応する、上記第２の３次元作業配列内の、Ｚ軸に垂直な複数の平面のそれぞれに対応する２次変換結果データ$c''_{kx,ky,jz}$が格納される。Ｘ方向の変換の終了後、プロセッサ間でのデータの転置（入れ替え）を行う。すなわち、今度は既にステップｃで述べたように、２次変換結果データの直方体を図８のようにＹ軸に垂直にスライスし、こうしてできる各面を一つのプロセッサに割り当てる（ステップ５８）。既にステップｃで述べたように、この割り当てに従い、各プロセッサが自分以外の全プロセッサとの間でそれぞれのプロセッサが生成した２次変換結果データの交換を行う。具体的には、この転置時には、各プロセッサは、上記第２の作業配列に、そのプロセッサに割り当てられた座標k_yの値を有するＹ軸に垂直な平面に属すべき、k_yが特定値で、k_x，j_zが種々の値を持つ２次変換結果データ$c''_{kx,ky,jz}$（但し、k_x＝０～ＮＸ－１，k_y＝特定値，j_z＝０～ＮＺ－１）を受信するように、プロセッサ間で２次変換結果データ$c''_{kx,ky,jz}$を交換する。転置の終了後、Ｚ方向の変換を行う（ステップ５９）。すなわち、各プロセッサは、既にステップｄで記載したように、そのプロセッサに新たに割り当てられた２次変換結果データに対して、式７により指定される変換を実行し、最終的な３次元のフーリエ変換結果データを生成する。転置により各ＸＺ平面は１台のプロセッサに担当されているから、この変換処理も通信なしに各プロセッサで独立に行える。具体的には、各プロセッサでは、Ｚ方向の変換は、k_y座標が特定の値を有し、k_x座標とＺ座標とがいろいろの値を有する一群の２次変換結果データ$c''_{kx,ky,jz}$に対して、式７により行なわれる。このとき、k_y座標が特定の値を有し、k_x座標が異なる一群の入力データに対してＺ方向の変換が高速フーリエ変換アルゴリズム（ＦＦＴ）を用いて実行される。この変換は、上記第２の３次元作業配列を使用して実行される。この変換の実行にあっては、k_z座標が異なる一群のデータに対して、プロセッサ内のベクトル演算器が使用され、計算がパイプライン的に実行される。その結果得られる最終フーリエ変換結果データ$c_{kx,ky,kz}$（但し、k_x＝０～ＮＸ－１，k_y＝特定値，k_z＝０～ＮＺ－１）は、第３の３次元作業配列の、これらの座標値k_x，k_y，k_zに対応するインデックスのところに格納される。したがって、図８のように、一つのプロセッサに一つの座標k_yを有す一つの平面が割り当てられた場合、各プロセッサでは、最終フーリエ変換結果データ$c_{kx,ky,kz}$は、上記第３の３次元作業配列の、特定の座標k_yを有する一つの平面に格納される。もし、図８において、k_y軸に垂直な複数の平面がそのプロセッサに割り当てられているときには、それぞれの平面に対応する、上記第３の３次元作業配列内の、k_y軸に垂直な複数の平面のそれぞれに対応する最終フーリエ変換結果データ$c_{kx,ky,kz}$が格納される。Ｚ方向の変換が終了すると、一次元の変換対象データf_jのフーリエ変換が終了し、重ね合わせの係数c_kが求まる。c_kの並び方は、原点からまずＹ方向に、Ｙ方向にＮＹ個行ったら次はＸ座標が１だけ増え、ＸＹ平面上にＮＸ＊ＮＹ個のデータが並んだら次はＺ座標が１だけ増える、という順で並ぶ（図３）。このデータの並び方と図８のデータの分割形式とを照らし合わせることにより、本実施の形態では、出力データc_kもサイクリック分割になっていることがわかる。上記第３の３次元作業配列内でも、最終フーリエ変換結果データ$c_{kx,ky,kz}$はこの並びに対応する並びを有する。ライブラリはこのデータ$c_{kx,ky,kz}$を一次元座標ｋの順に並び替えて一次元配列Ｆに格納し（ステップ６１）、リターンする（ステップ６２）。本ライブラリでは、従来法で必要であった変換後のデータ分割形式

の変更が不要となり、通信の削減により従来法を上回る並列化効率を得ることが可能となり、フーリエ変換時間を低減できる。なお、ＮＸ，ＮＹ，ＮＺの決め方としては、プロセッサ台数をｐとすると、Ｙ方向、Ｘ方向の変換でＺ方向に垂直な面でデータを分割することから、ＮＺ≧ｐが成り立つ必要がある。また、Ｚ方向の変換ではＹ方向に垂直な面でデータを分割することから、ＮＹ≧ｐも成り立つ必要がある。また、並列計算機２８の各プロセッサ２９がベクトル演算器（図示せず）を備えると仮定した。このような並列計算機では、このベクトル演算器を効率的に使うためには、ベクトル化の対象となるループの長さ（すなわち、同じ演算を受けるデータ群（ベクトルデータ）の要素数であり、ベクトル長とも言われる）をできるだけ長く取る必要があることが知られている。本アルゴリズムで式５から７を計算するときには、このベクトル演算器が使用される。ベクトル化の対象となるループは、フーリエ変換にも並列化にも使わない座標軸の方向で複数のデータに対して同じ演算を実行する計算であり、Ｙ方向、Ｘ方向、Ｚ方向の変換において、それぞれＸ方向、Ｙ方向、Ｘ方向での演算となる。したがって、ベクトル演算器の性能を引き出すには、ＮＹ≧ｐ，ＮＺ≧ｐの２つの条件を満たしつつＮＸとＮＹをできるだけ大きく取るようにＮＸ，ＮＹ，ＮＺを決めることが望ましい。なお、並列計算機２８は、ベクトル演算器を有すると仮定したいが、この演算器がフーリエ変換において必要な全ての演算の一部の演算をパイプライン的に実行できるものでもよい。さらに、並列計算機２８がベクトル演算器を有しない並列計算機であっても、ループ長を大きくすることが高速化に有効である場合が多い。また、以上の動作の説明では、並列計算機２８がメモリ２９と演算器（図示せず）の間に複数のベクトルレジスタを有しないと仮定し、各方向の変換で利用される配列はメモリ２９から直接演算器に読み出され、あるいはその変換で生成される配列はメモリ２９に直接演算器から書き込まれるかのように説明した。しかし、複数のベクトルレジスタを有する並列計算機では、メモリ２９上の配列に対する演算あるいはその演算の結果得られた配列のメモリ２９への格納は、これらのレジスタを介して実行させればよいことは当業者に明らかである。本実施の形態では本ライブラリは逆フーリエ変換を実行するためのプログラムを有しない。後で説明するように、シミュレーションプログラムが逆フーリエ変換を必要とするときには、シミュレーションプログラムの方で、逆フーリエ変換の対象のデータの複素共役データを生成し、そのデータに対してフーリエ変換を本ライブラリに要求する。この複素共役データに対して得られたフーリエ変換データの複素共役をシミュレーションプログラムが生成する。しかし、本ライブラリにフーリエ変換の機能を持たせることもできる。すなわち、本ライブラリの引数としてフーリエ変換か逆フーリエ変換かを指定する引数を追加し、シミュレーションプログラムが逆フーリエ変換を要求したときには、本ライブラリで、変換対象データの複素共役を求め、これにフーリエ変換を上記のようにして実行し、得られた結果データの複素共役を求め、それを逆フーリエ変換結果データとしてシミュレーションプログラムに戻せばよい。

（５）シミュレーションプログラム

本実施の形態において使用するシミュレーションプログラムの例として気象計算のための並列プログラムを図１１に示す。気象計算は本来３次元の計算であるが、現在は計算機能力の制約から２次元で行うことも多い。そこで本実施の形態では、２次元の気象予測対象とする領域（これが計算対象領域となる）の場合を例にとってシミュレーションプログラムを説明する。ユーザは、予め全計算対象領域を複数の部分計算領域に区分し、それぞれをいずれか一つのプロセッサに割り当てる。さらに、各部分計算領域のサイズＮ１，Ｎ２、フーリエ変換で用いるＮＸ，ＮＹ，ＮＺなどのパラメータを指定する。ユーザが本プログラムを並列計算機２８で使用するときには、まず、ワークステーション１が、並列計算機２８内の特定のプロセッサ（たとえばプロセッサ０）と交信して、このプログラムと上記ユーザ指定の情報と、空気の熱伝導率などの計算に使用する物質定数と、全計算対象領域に内の観測によって得られた温度・風速・圧力などの初期値データとを、その特定のプロセッサ０を介して外部記憶装置３１に記憶する。その後、その特定のプロセッサ０が、各プロセッサに本プログラムをロードし、全プロセッサで本プログラムを起動する。本プログラムは、並列計算機２８内の全プロセッサで全く同じようにして並行して実行される。本プログラムでは、起動されると、まず計算領域のサイズＮ１，Ｎ２、フーリエ変換で用いるＮＸ，ＮＹ，ＮＺ、空気の熱伝導率などの物質定数などのパラメータと、観測によって得られた温度・風速・圧力などの初期値データとを外部記憶装置３１から入力する（ステップ３２）。本プログラムは、それがロードされたプロセッサがどの部分計算領域に関する計算を実行するかを判断するようにプログラムされていると仮定する。このステップでは、各プロセッサは、プロセッサに依らないで使用される上記パラメータを入力するとともに、外部記憶装置３１に記憶された全計算対象領域に対する初期値データの内、そのプロセッサに割り当てられた部分計算領域に関する初期値データを選択して外部記憶装置３１から入力する。なお、上記（３）の「多次元フーリエ変換への応用」の項で述べたように、本発明の方式による２次元高速フーリエ変換では、入力データが第１の座標方向にサイクリック分割されている必要がある。すなわち、サイズＮ１×Ｎ２のメッシュ上で定義されたある物理量$A_{j1,j2}$（ただし$j_1=0, 1, \ldots, N1-1$，$j_2=0, 1, \ldots, N2-1$）のうち，要素$A_{m*NPU+i,j2}$（$m=0, 1, \ldots, N1/NPU$

－１，ｉ＝０，１，...，ＮＰＵ－１，　j_2＝０，１，...，Ｎ２－１)は第ｉ番目のプロセッサに割り当てられている必要がある。そこで，本実施の形態のシミュレーションプログラムでも，２次元高速フーリエ変換での入力形式に合わせて，このように第１の座標方向をサイクリック分割することによって得られる部分計算領域を用いる。その後計算に必要な前処理を行う（ステップ３３）。ここで前処理とは、観測によって得られた温度・風速・圧力などのデータに対して補間を行い、計算に必要なメッシュポイントでの温度・風速・圧力などのデータを得ることである。これらの処理が終わった後、以下に説明する繰り返しループにより各時間ステップでの温度・風速・圧力などの量を順々に求めていく。基礎となる方程式は、以下に示す風速に対する運動方程式、質量保存の式、温度変化を表す式の３本である。

【数１２】

$$du/dt=-2\Omega\times u-(1/\rho)\nabla p+F_u \quad ,\ ...\ (12)$$

【数１３】

$$d\rho/dt=-\rho\nabla\cdot u \quad ...\ (13)$$

【数１４】

$$dT/dt=-\kappa\nabla^2T+u\cdot\nabla T \quad ...\ (14)$$

ここで、ｕは風速、ｐは圧力、Ｔは温度を表し、Ωはコリオリ力と呼ばれる地球の自転による力、F_uはそれ以外の外力、ρは空気の密度、κは空気の熱伝導率を表す。これらの式から次の時刻でのデータの値を求めるには、まずフーリエ変換により格子点上の温度Ｔ、圧力ｐおよび風速ｕをそれぞれ波数空間でのデータに変換する。そのために、それぞれの物理量についてのデータについて２次元高速フーリエ変換ライブラリＦＦＴ２Ｄを順次コールする（ステップ３４）。ライブラリＦＦＴ２Ｄのコール時には、既に述べた引数を指定する。波数空間でそれぞれの物理量のデータを微分する（ステップ３５）。すなわち、ライブラリＦＦＴ２Ｄから与えられる、各物理量に関するフーリエ変換係数データを波数空間で微分し、その物理量についての、波数空間の格子点上での温度勾配∇Ｔ、２次微分∇^2Ｔ、圧力勾配∇ｐ、風速の発散$\nabla\cdot$ｕ等の微分に関連するデータを求める。各物理量についての上記微分に関連するデータを逆フーリエ変換して、実空間の格子点上での温度勾配∇Ｔ、２次微分∇^2Ｔ、圧力勾配∇ｐ、風速の発散$\nabla\cdot$ｕ等の微分に関連するデータを求める（ステップ３６）。逆フーリエ変換するには、すでに述べた式８、９を使用する。すなわち、各物理量についての上記微分後のデータの複素共役なデータを生成し、ライブラリＦＦＴ２Ｄをコールしてこの複素共役なデータに対するフーリエ変換を要求する。さらに、得られたフーリエ変換結果データの複素共役なデータを生成し、この生成された複素共役なフーリエ変換結果データを逆フーリエ変換結果データとして使用する。この後、上記逆フーリエ変換で得られた微分に関連するデータを、式１２－１４の右辺に代入し、風速ｕ、空気密度ρ、温度Ｔのそれぞれに関する時間微分を決定し、得られたそれらの時間微分を用いて、次の時間ステップでの温度・風速・圧力を求める（ステップ３７）。なお、ステップ３４，３５，３６で、フーリエ変換により実空間上の格子点のデータを波数空間上のデータに変換してそこで微分関連データを求め、得られた微分関連データを逆フーリエ変換して実空間に関する微分関連データを得るのは、その方が微分が精度良く計算できるからであり、本シミュレーションプログラムでは、この計算部分で２次元フーリエ変換ライブラリＦＦＴ２Ｄを用いる。上記のループでは、各時間ステップ毎に、求める時刻までの計算が終了したかどうかを判定し（ステップ３８）、終了したら、後処理を行い（ステップ３９）、予測結果データとして出力する（ステップ４０）。後処理では、主に計算を行うメッシュポイントと予測結果データが必要な点とがずれている場合に、計算結果データを補間して必要な点での予測値を計算するなどの処理を行う。出力処理４０では、各プロセッサは、生成したデータを外部記憶装置３１に書き戻し、上記特定のプロセッサはシミュレーションプログラムの実行終了時に、このデータをワークステーション１に一つの結果データとして転送する。なお、以上ではシミュレーションプログラムは、並列計算機２８内の全プロセッサで全く同じようにして並行して実行されると仮定した。さらに、それがロードされたプロセッサがどの部分計算領域に関する計算を実行するかを判断するようにプログラムされていると仮定する。しかし、本発明に依るフーリエ変換を利用するプログラムはこのようなプログラムに限定されないことはいうまでもない。上記フーリエ変換ライブラリＦＦＴ２Ｄを使用するには、それぞれのライブラリが要求する上記複数の引数を指定することが必要であり、それらの引数の生成あるいは獲得は他の方法でも良い。たとえば、本プログラムは、並列計算機２８内の特定の一つのプロセッサで実行される単独処理部分と全プロセッサで並行して実行される並列処理部分とから構成されてもよい。たとえば、本プログラムがいずれかのプロセッサで起動されたときに、そのプロセッサが上記特定の一つのプロセッサであるときにその単独処理部分が実行され、そうでないときには上記並列処理部分のみが実行される。上記単独処理部分では、各プロセッサが担当する部分計算領域を判断し、その結果を他のプロセッサに通知するように構成できる。上記の例では気象予測計算を行う場合を例にとって説明したが、本発明の手法は、これ以外の応用例についても、並列計算機上で高速フーリエ変換を用いてシミュレーションを行う場合

に適用できることは明らかである。一次元フーリエ変換ライブラリＦＦＴ１Ｄについても全く同様である。

＜発明の実施の形態２＞上記の実施の形態１では、フーリエ変換ライブラリは、図９の方式２を採用した。しかし、本実施の形態では、フーリエ変換ライブラリは、図９の方式３を採用する。この方式３では、Ｙ方向の変換を行った後で転置を行い、その後Ｘ方向とＺ方向の変換を行う。この方式２では、フーリエ変換は具体的には以下のようにして実行される。

（ステップａ’）Ｙ方向の変換

Ｙ方向の変換は、従来の転置アルゴリズムについて既に述べたステップ１あるいはａ’の要領で実行される。既に述べたごとく、フーリエ変換対象データのデータ分割は、サイクリック分割である。

（ステップｂ’）データ転置

式６によるＸ方向の変換を行うには、３次元波数空間の座標k_yの特定の値と、３次元実空間の座標（j_x，j_z）の全ての値に対して得られた１次変換結果データ$c'_{jx,ky,jz}$が必要である。方式３では、方式２と異なり、Ｙ方向の変換で得られた１次変換結果データ$c'_{jx,ky,jz}$は、Ｙ軸に垂直な複数の平面で分割される。このことは、図８に示すように、３次元波数空間を、そのk_y軸に垂直な複数の平面で分割して、分割後の部分空間（上記複数の平面）の各々を一つのプロセッサに割り当てることを意味する。すなわち、各プロセッサへの座標k_yの特定の値を割り当てる。具体的には、k_y＝０，１，２，３，，，の１変換結果データ$c'_{jx,ky,jz}$を順次プロセッサ０，１，２，，，７に割り当てる。この割り当てに従い、後にＸ方向の変換を実行するに必要なデータをプロセッサ間で転送する処理がここのステップｂ’で実行される。各プロセッサが、そのプロセッサに割り当てられた３次元波数空間の座標k_yの特定の値と、３次元実空間の座標（j_x，j_z）の全ての値とに対して得られた全ての１次変換結果データ$c'_{jx,ky,jz}$を使用してＸ方向の変換を実行できるように、全プロセッサ間で１次変換結果データの転送が行われる。すなわち、各プロセッサは、３次元実空間の座標（j_x，j_z）の全て値と、３次元波数空間の座標k_yの特定の値との組に対して得られた１次変換結果データ$c'_{jx,ky,jz}$（但し、j_x＝０～（ＮＸ－１）、j_z＝０～（ＮＺ－１）、k_y＝特定値）の内、自プロセッサが生成しなかったデータを他のプロセッサから受信するように、全プロセッサの間で２次変換結果データを転送する。

（ステップｃ’）Ｘ方向の変換

次に、Ｘ方向の変換を実行する。各プロセッサは、式６により、他のプロセッサとの通信をしないで、そのプロセッサに割り当てられた３次元波数空間の座標k_yの特定の値と、３次元波数空間の座標k_xの全ての値と、３次元実空間の座標j_zの全ての値に関連するＮＸ＊ＮＺ個の２次変換結果データ（$c''_{kx,ky,jz}$）（k_x＝０～ＮＸ－１、j_z＝０～ＮＺ－１）を得る。

（ステップｄ’）Ｚ方向の変換

各プロセッサは、そのプロセッサに割り当てられた３次元波数空間の座標k_yの特定の値と、３次元波数空間の他の二つの座標k_x、k_zの全ての値を有する最終的なフーリエ変換結果データ$c_{kx,ky,kz}$（但し、k_x＝０～ＮＸ－１、k_y＝特定値、k_z＝０～ＮＺ－１）を計算する。各プロセッサはこの計算を互いに並列に実行できる。この結果、座標k_y＝０，１，２，３，，，に対応するフーリエ変換係数c_0，c_1，c_2，，，が、順次プロセッサ０，１，２，，，で生成され、全フーリエ変換結果データ$c_{kx,ky,kz}$は、プロセッサ間でサイクリックに分割されていることが分かる。本方式は、実施の形態１で使用した方式２に比べ、複素指数関数テーブルを格納する配列ＢＴの容量の点で有利となる。実際、式６により、Ｘ方向の変換における複素指数関数の値はＸ方向、Ｙ方向のインデックスのみに依存し、Ｚ方向のインデックスには依存しない。したがって、実施の形態１のようにＸ方向の変換においてＺ軸に垂直な分割を採用した場合には、各プロセッサが同じテーブルを重複して持つことになる。それに対して本方式では、分割をＹ軸に垂直な面で行うので、各プロセッサが自分の計算に必要なテーブルの一部分のみを持つことになり、重複はない。これにより、本方式ではＸ方向の変換に必要なテーブルの大きさが１／（プロセッサ台数）に削減できる。本実施の形態では、ベクトル化の対象となるループはＹ方向、Ｘ方向、Ｚ方向の変換において、それぞれＸ方向、Ｚ方向、Ｘ方向となるので、ベクトル並列計算機の性能を引き出すには、ＮＹ≧ｐ，ＮＺ≧ｐの２つの条件を満たしつつＮＸとＮＺをできるだけ大きく取るのがよい。

＜発明の実施の形態３＞本実施の形態において対象となる並列計算機は、実施の形態１で説明した図１の並列計算機システムとほぼ同様であるが、各プロセッサは同一のベクトル演算器を内蔵し、かつ、外部記憶装置中３２に、そのベクトル演算器の性能に関するデータベースを持つ。ベクトル演算器の演算性能とはたとえば単位時間あたりに実行可能な演算回数である。ベクトル演算器性能データベース中には、ベクトル演算器の性能データがループ長Ｌの関数（ｇ（Ｌ））の形で格納されている。実施の形態１では、フーリエ変換のためのパラメータＮＸ，ＮＹ，ＮＺはプログラムへの入力として決定していたが、これを並列計算機２８の各々のプロセッサを構成するベクトル演算器の特性に応じて最適化することにより、さらに効率的な計算が可能となる。一般に、ベクトル演算器の演算性能は、ループ長Ｌの関数ｇ（Ｌ）である。ｇ（Ｌ）は通常、Ｌに対して単調に増加する関数である。いま、実施の形態１のフーリエ変換方式でＹ方向の変換を計算するステップの演算量を考えると、ＮＹ次のフーリエ変換を一回行うための演算量はＮＹｌｏｇＮ

Ｙであり、これをＮＸ＊ＮＺ組計算するから、全体での演算量はＮＸ＊ＮＹ＊ＮＺｌｏｇＮＹ＝Ｎｌｏｇ ＮＹである。同様にして、ＮＸ方向、ＮＺ方向での演算量は、それぞれＮｌｏｇＮＸ、ＮｌｏｇＮＺである。一方、それぞれの演算におけるベクトル化のループ長は、実施の形態１で述べたようにＮＸ，ＮＹ，ＮＸであるから、ベクトル演算器の演算性能はそれぞれｇ（ＮＸ），ｇ（ＮＹ），ｇ（ＮＸ）となる。演算時間ｔは演算量を演算性能で割ることによって得られ、合計で

t=NlogNY/g(NX)+NlogNX/g(NY)+NlogNZ/g(NX)

となる。したがって、プロセッサ台数をｐとするとき、ＮＸ≧ｐ，ＮＺ≧ｐという条件の下でｔを最小化するようにＮＸ，ＮＹ，ＮＺを決めることにより、ベクトル演算器の性能を最大限に発揮できる高速フーリエ変換が実現できる。本実施の形態でのライブラリのフローチャートを図１２に示す。処理は、ＮＸ，ＮＹ，ＮＺの決定部分（ステップ４３）を除いては、実施の形態１（図１０）と同様である。ステップ４３では、上記ベクトル演算器性能データベースを用いて、ＮＸ≧ｐ，ＮＺ≧ｐという条件の下で上記演算時間ｔを最小化するようにＮＸ，ＮＹ，ＮＺを決める。その後の処理は、実施の形態１と同様である。このライブラリへのコール文ではシミュレーションプログラムはこれらのパラメータＮＸ，ＮＹ，ＮＺを指定する必要はない。本実施の形態の方式によれば、ユーザは自分でＮＸ，ＮＹ，ＮＺを計算することなく、ベクトル演算器を内蔵する並列計算機の性能を最大限に引き出すことが可能となる。なお、ＮとＮＰＵとが一般の整数の場合には、ＮＸ，ＮＹ，ＮＺを変えることにより、入力データのプロセッサへの分割形式も変更する必要があるが、フーリエ変換でもっともよく利用される、ＮおよびＮＰＵが共に２のべき乗の場合には、ＮＸ，ＮＹ，ＮＺを変えても、分割形式を変更する必要がない場合がある。実際、２つの組（ＮＸ，ＮＹ，ＮＺ）＝（ＮＸ１，ＮＹ１，ＮＺ１）、（ＮＸ２，ＮＹ２，ＮＺ２）が共にＮＹ≧ＮＰＵ，ＮＺ≧ＮＰＵの２つの条件を満たしているとき、入力データf_jを図２に示す順番で直方体状に並べ、これをＺ軸に垂直な面でスライスして、各面をサイクリックにプロセッサ０，１，．．．，ＮＰＵ－１に割り当てたとする。すると、（ＮＸ，ＮＹ，ＮＺ）＝（ＮＸ１，ＮＹ１，ＮＺ１）の場合は、f_jの属する面は上からＭＯＤ（f_j，ＮＺ１）＋１番目であり、この面を担当するプロセッサの番号は

MOD(MOD(f_j,NZ1),NPU)

である。一方、（ＮＸ，ＮＹ，ＮＺ）＝（ＮＸ２，ＮＹ２，ＮＺ２）の場合も同様にして、f_jを担当するプロセッサの番号は

MOD(MOD(f_j,NZ2),NPU)

となる。ところが、いまＮＺ１≧ＮＰＵ，ＮＺ２≧ＮＰＵであり、ＮＺ１，ＮＺ２，ＮＰＵはすべて２のべき乗であるから、ＮＺ１，ＮＺ２は共にＮＰＵの倍数である、したがって、

MOD(MOD(f_j,NZ1),NPU)

=MOD(MOD(f_j,NZ2),NPU)

=MOD(f_j,NPU)

すなわち、f_jを担当するプロセッサの番号は、どちらの場合も同じである。以上の考察より、ＮおよびＮＰＵが共に２のべき乗で、ＮＹ≧ＮＰＵ，ＮＺ≧ＮＰＵの２つの条件が成り立っている限り、ＮＸ，ＮＹ，ＮＺを変えても、入力データf_jのプロセッサへの分割形式は変更する必要がないことがわかる。このことを利用し、分割形式を変えずに済む範囲でＮＸ，ＮＹ，ＮＺの最適化を行えば、分割形式変更に伴う新たな通信オーバーヘッドを生じることなく、ベクトル演算器を含む並列計算機での処理速度、具体的には、フーリエ変換速度を向上させることができる。

＜発明の実施の形態４＞本発明による高速フーリエ変換を用いてシミュレーションを行う他の例として、半導体デバイス等における電子構造計算を説明する。電子構造計算は、その結果を利用して半導体デバイスの設計、とくにデバイス構造の決定に使用されている。電子構造計算では、２次元または３次元のメッシュで定義された電子の波動関数ｕ（ｒ）を、次のシュレディンガー方程式

【数１５】

$$du(r)/dt=-(h^2/2m)\nabla^2u(r)+(E-V(r))u(r) \quad \cdots \quad (15)$$

に従って計算することにより、半導体の性質を決定するバンドギャップの大きさや、結晶の構造安定性などを求める。ただし、上式で、ｈはプランク定数、ｍは電子の質量、Ｅは対象とする波動関数のエネルギーレベル、Ｖは結晶中の原子や他の電子によるポテンシャルエネルギーを表す。式１５の計算では、波動関数ｕ（ｒ）の２次微分$\nabla^2u(r)$が必要であるが、気象計算の例において述べたのと同様な理由により、この部分はｕ（ｒ）をフーリエ変換により波数空間に移してから計算し、結果を逆フーリエ変換で再び実空間に戻す。したがって、並列計算機上で電子構造計算を行う場合には、この部分で本発明の高速フーリエ変換方法が適用できる。

＜変形例＞本発明は、以上の実施の形態に限定されるのではなく、以下に例示する変更あるいは変形以外のいろいろの変更あるいは変形により実現可能である。

（１）本発明によるフーリエ変換方法は、シミュレーションに限らず他の用途にも使用できるのは言うまでもない。たとえば、伝送される信号あるいは地震波等の波動の解析に利用でき、解析の結果を用いて、信号伝送に関係する装置、例えば伝送装置あるいは伝送線路の設計を行うことができ、あるいは地震を利用した応用、例えば資源開発等にも利用できる。

（２）以上の実施の形態では、フーリエ変換変換はそのために用意されたフーリエ変換ライブラリにより実行さ

れた。しかし、本発明は、フーリエ変換を使用するアプリケーションプログラム自身にこのフーリエ変換手順を実行するプログラムを組み込んでもよいことは明らかである。このようなシミュレーションプログラムは。プログラムを磁気記憶装置のようなプログラム記録媒体に記憶して販売できる。

（３）本発明は、フーリエ変換対象データf_jが実データであるときにも適用できる。その場合に、Ｙ方向等の変換のときに、係数の計算においては、虚数部の計算を省略することができる。

以上から明らかなように、本発明によれば、並列計算機を使用してフーリエ変換を従来より高速に実行できる。たとえば本出願人により開発された並列計算機ＳＲ２２０１を用いて本発明の効果を評価した結果では以下の通りである。３次元フーリエ変換を実行する場合、従来法では、２５６×２５６×２５６のサイズのデータを２５６台のプロセッサを用いて変換するのに、約０．２６秒の時間が必要である。この内訳は、計算に０．１４秒、途中でのデータの転置に０．０６秒、最後のデータの並べ替えに０．０６秒の時間がかかる。実施の形態１あるいは２に記載の方法によれば、計算と転置の時間は従来法と同じであり、最後のデータの並べ替えが省略できるので、０．２０秒でフーリエ変換を行うことができ、約２４％の高速化が達成できる。とくに、実施の形態１で記載した気象計算を、３次元フーリエ変換を用いて行う場合、気象計算では全計算時間の約５０％がフーリエ変換で占められるため、約１２％の高速化が得られる。また、実施の形態４で記載した電子構造計算を３次元フーリエ変換を用いて行う場合、通常全実行時間の３０％程度がフーリエ変換で占められるため、約７％の高速化が達成できる。

【発明の効果】以上説明したように、本発明によれば、並列計算機上でのフーリエ変換を高速に実行できる。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態で使用する並列計算機の概略構成図。

【図２】本発明の第１の実施の形態で使用する一次元変換対象データの３次元データへの写像を説明する図。

【図３】本発明の第１の実施の形態で使用する一次元変換結果データの３次元データへの写像を説明する図。

【図４】本発明の第１の実施の形態で使用する、プロセッサへデータを割り当てる第１の方法を示す図。

【図５】従来技術で使用する、プロセッサへデータを割り当てる他の方法を示す図。

【図６】本発明の第１の実施の形態で使用する、一次元変換対象データのプロセッサ間データ分割形式を説明する図。

【図７】従来技術による、一次元変換結果データのプロセッサ間データ分割形式を説明する図。

【図８】本発明の第１の実施の形態で使用する、プロセッサへデータを割り当てる第２の方法を示す図。

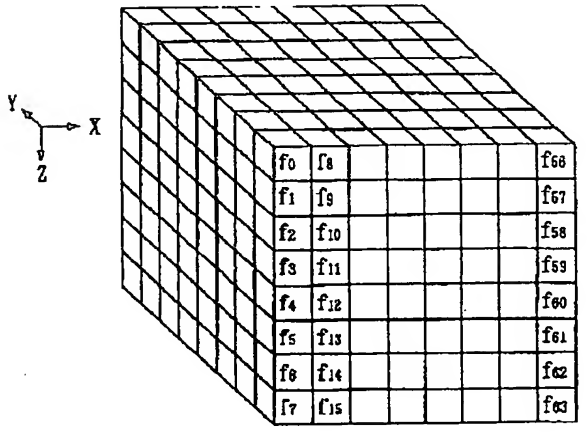
【図９】本発明に至る前に比較検討した、複数のフーリエ変換変換手順を示す図。

【図１０】本発明の実施の形態１で使用するフーリエ変換ライブラリのフローチャート。

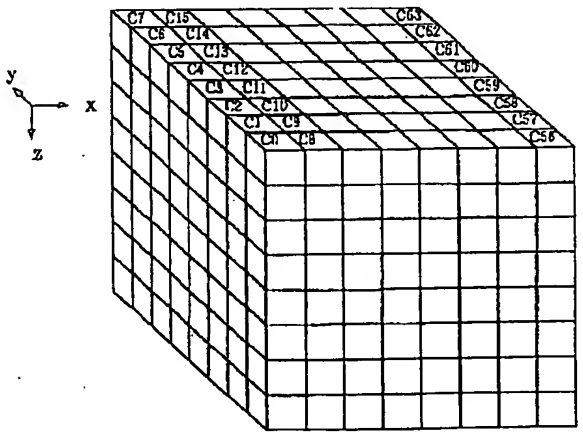
【図１１】本発明の実施の形態１で使用するシミュレーションプログラムのフローチャート。

【図１２】本発明の実施の形態３で使用するフーリエ変換ライブラリのフローチャート。

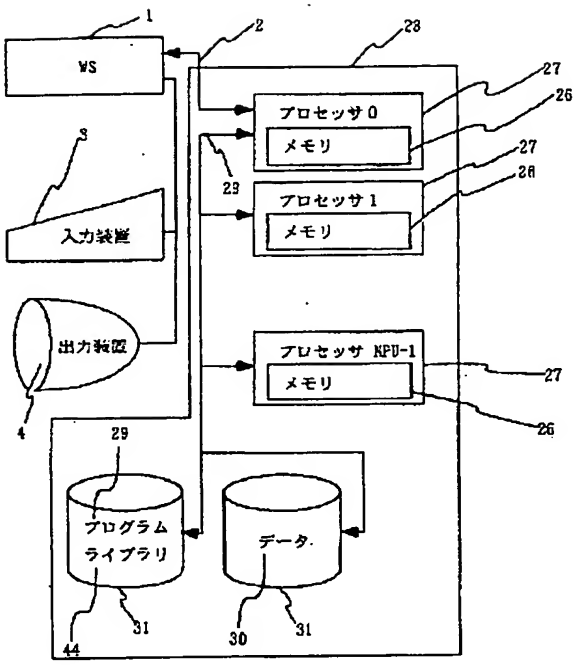
【図２】



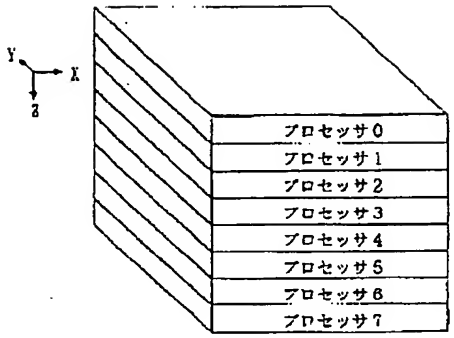
【図３】



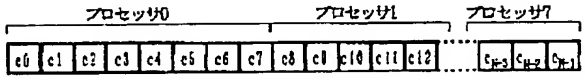
【図１】



【図４】



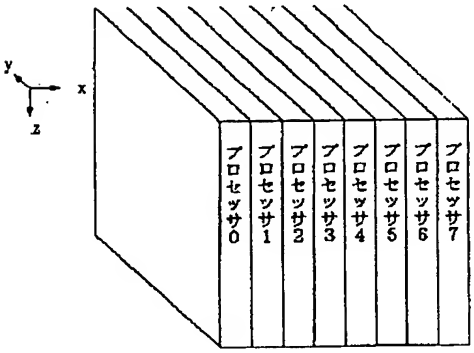
【図７】

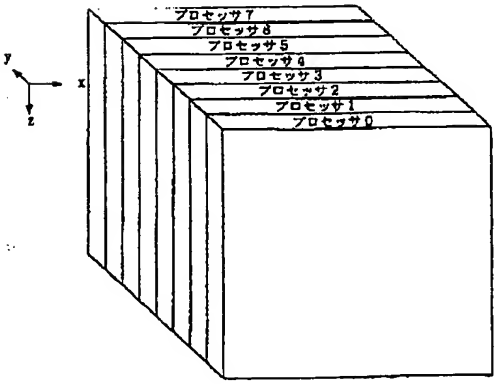


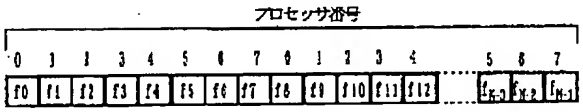
【図５】

【図６】

【図８】



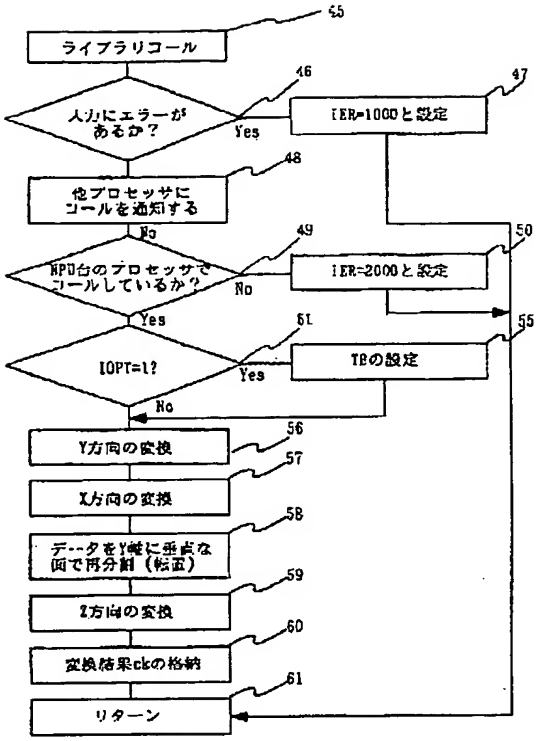




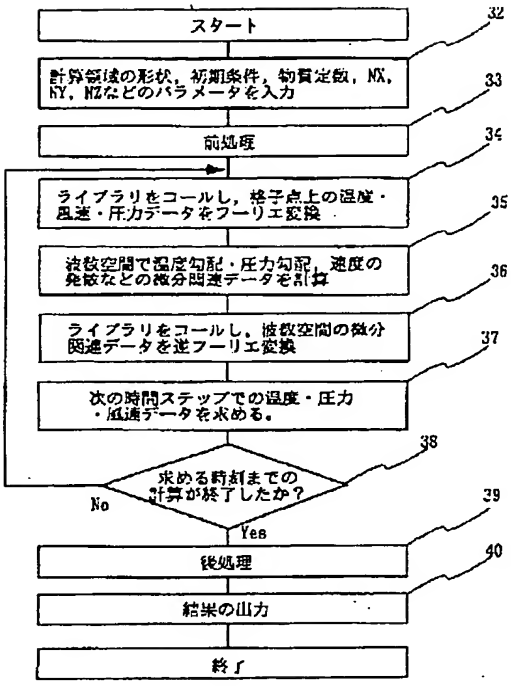
【図９】

変換方向	Y	X	Z
方式1	Z軸に垂直に分割	Z軸に垂直に分割	X軸に垂直に分割
方式2	Z軸に垂直に分割	Z軸に垂直に分割	Y軸に垂直に分割
方式3	Z軸に垂直に分割	Y軸に垂直に分割	Y軸に垂直に分割
方式4	X軸に垂直に分割	Y軸に垂直に分割	Y軸に垂直に分割

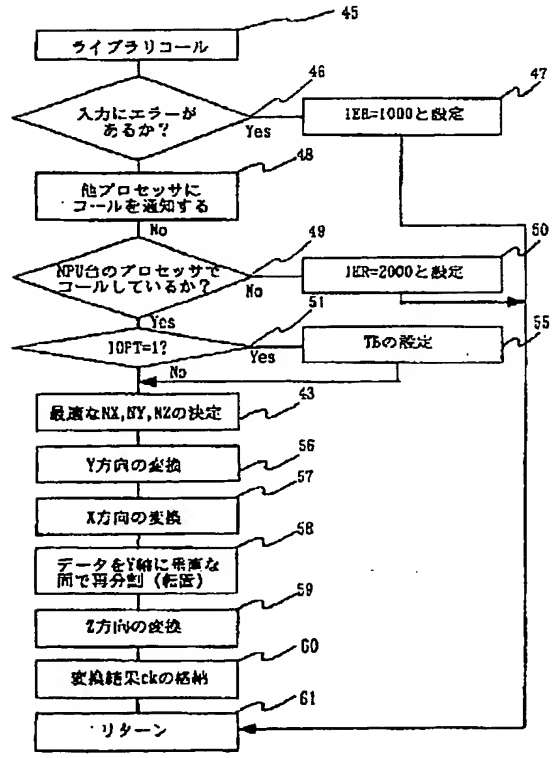
【図１０】


ライブラリコール
45
入力にエラーがあるか？
46
Yes
IER=1000と設定
47
他プロセッサにコールを通知する
48
No
NPU台のプロセッサでコールしているか？
49
No
IER=2000と設定
50
Yes
IOPT=1?
51
Yes
TBの設定
55
No
Y方向の変換
56
X方向の変換
57
データをY軸に垂直な面で再分割（転置）
58
Z方向の変換
59
変換結果ckの格納
60
リターン
61

【図１１】


スタート
計算領域の形状，初期条件，物質定数，NX，NY，NZなどのパラメータを入力
32
前処理
33
ライブラリをコールし，格子点上の温度・風速・圧力データをフーリエ変換
34
波数空間で温度勾配・圧力勾配，速度の発散などの微分関連データを計算
35
ライブラリをコールし，波数空間の微分関連データを逆フーリエ変換
36
次の時間ステップでの温度・圧力・風速データを求める。
37
求める時刻までの計算が終了したか？
38
No
Yes
後処理
39
結果の出力
40
終了

【図１２】


ライブラリコール
45
入力にエラーがあるか？
46
Yes
IER=1000と設定
47
他プロセッサにコールを通知する
48
No
NPU台のプロセッサでコールしているか？
49
No
IER=2000と設定
50
Yes
IOPT=1?
51
Yes
TBの設定
55
No
最適なNX,NY,NZの決定
43
Y方向の変換
56
X方向の変換
57
データをY軸に垂直な面で再分割（転置）
58
Z方向の変換
59
変換結果ckの格納
60
リターン
61